

References

- [1] S. Haykin, *Adaptive filter theory*, Prentice Hall, Englewood Cliffs, 2nd Ed., 1991.
- [2] E. Frantzeskakis and K.J.R. Liu, "A Class of Square Root and Division Free Algorithms and Architectures for QRD-Based Adaptive Signal Processing", *IEEE Trans. on Signal Processing*, vol 42, no 9, pp.2455-2469, September 1994.
- [3] K.J.R. Liu, S.F. Hsieh, and K. Yao, "Systolic Block Householder Transformation for RLS Algorithm with Two-Level Pipelined Implementation", *IEEE Trans. on Signal Processing*, Vol 40, No 4, pp.946-958, April 1992.
- [4] A. Lee, "Centrohermitian and skew-centrohermitian matrices", *Linear Algebra and its Applications*, vol. 29, pp. 205-210, 1980.
- [5] S.L. Marple, Jr, *Digital spectral analysis with applications*, Prentice-Hall, Englewood Cliffs, New Jersey, 1987
- [6] A. Klein and P.W. Baier, "Linear unbiased data estimation in mobile radio systems applying CDMA", *IEEE Journal on Selected Areas in Communications*, vol. 11, pp. 1058-1066, 1993.



Continuing with the example of a transversal filter, suppose we are given a *set of observations* represented by the tap-input vector $\mathbf{u}(i)$, where the time index $i = 1, 2, \dots, n$. We assume that the tap-weight vector \mathbf{w} of the transversal filter is held *constant* for the *entire* observation interval. At time i , we define the estimation error as

$$e(i) = d(i) - \mathbf{w}^H \mathbf{u}(i) \quad (1.20)$$

where $d(i)$ is the corresponding value of the desired response. In this case, we may define the cost function as

$$\begin{aligned} \mathcal{E}(\mathbf{w}, n) &= \sum_{i=1}^n |e(i)|^2 \\ &= \sum_{i=1}^n |d(i) - \mathbf{w}^H \mathbf{u}(i)|^2 \end{aligned} \quad (1.21)$$

Note that whereas the ensemble-averaged cost function $J(\mathbf{w})$ in Eq. (1.19) depends only on the tap-weight vector \mathbf{w} , the time-averaged cost function $\mathcal{E}(\mathbf{w}, n)$ depends on \mathbf{w} and the observation interval n . Accordingly, the minimization of $\mathcal{E}(\mathbf{w}, n)$ with respect to \mathbf{w} yields a solution for the tap-weight vector that varies with the observation interval. This second approach forms the basis of the *method of least squares*.

The cost functions $J(\mathbf{w})$ and $\mathcal{E}(\mathbf{w}, n)$ are both *convex* with a unique *minimum* point. Accordingly, their use yields a unique solution for the tap-weight vector of the transversal filter. A qualification in the context of the method of least squares, however, is in order. In particular, in Eq. (1.21) it is assumed that the number of observations n is larger than the number of tap weights constituting the vector \mathbf{w} ; that is, we have an *overdetermined* system with more equations than unknowns.

A limitation of second-order statistics (e.g., the mean-square-error criterion) is that they are *phase blind*. We may overcome this limitation by the use of a *non-linear* cost function. By so doing, the filter is enabled to extract information (particularly phase information) from the input signal in a more efficient manner. For this to be possible, however, the input signal must have *non-Gaussian* statistics. The use of such an approach provides the basis for an important class of *nonlinear* adaptive filtering algorithms that can perform *blind deconvolution*, blind in the sense that the algorithms do *not* require a desired response. The issue of blind deconvolution (and its use in blind equalization) are covered in Chapter 20.

1.7 APPLICATIONS

The desirable features of an adaptive filter, namely, the ability to operate satisfactorily in an unknown environment and also track time variations of input statistics, make the adaptive filter a powerful device for signal-processing and control applications.

Indeed, adaptive filtering has been successfully applied in such diverse fields as communications, radar, sonar, seismology, and biomedical engineering. Although these applications are indeed quite different in nature, nevertheless, they have one basic common feature: An input vector and a desired response are used to compute

an estimation error, which is in turn used to control the values of a set of adjustable filter coefficients. The adjustable coefficients may take the form of tap weights, reflection coefficients, or rotation parameters, depending on the filter structure employed. However, the essential difference between the various applications of adaptive filtering arises in the manner in which the desired response is extracted. In this context, we may distinguish four basic classes of adaptive filtering applications, as depicted in Fig. 1.7. For convenience of presentation, the following notations are used in this figure:

u = input applied to the adaptive filter

y = output of the adaptive filter

d = desired response

$e = d - y$ = estimation error.

The functions of the four basic classes of adaptive filtering applications depicted herein are as follows:

- I. *Identification* [Fig. 1.7(a)]. The notion of a *mathematical model* is fundamental to sciences and engineering. In the class of applications dealing with identification, an adaptive filter is used to provide a linear model that represents the best fit (in some sense) to an *unknown plant*. The plant and the adaptive filter are driven by the same input. The plant output supplies the desired response for the adaptive filter. If the plant is dynamic in nature, the model will be time varying.
- II. *Inverse modeling* [Fig. 1.7(b)]. In this second class of applications, the function of the adaptive filter is to provide an *inverse model* that represents the best fit (in some sense) to an *unknown noisy plant*. Ideally, the inverse model has a transfer function equal to the *reciprocal (inverse)* of the plant's transfer function. A delayed version of the plant (system) input constitutes the desired response for the adaptive filter. In some applications, the plant input is used without delay as the desired response.
- III. *Prediction* [Fig. 1.7(c)]. Here the function of the adaptive filter is to provide the best *prediction* (in some sense) of the present value of a random signal. The present value of the signal thus serves the purpose of a desired response for the adaptive filter. Past values of the signal supply the input applied to the adaptive filter. Depending on the application of interest, the adaptive filter output or the estimation (prediction) error may serve as the system output. In the first case, the system operates as a *predictor*; in the latter case, it operates as a *prediction-error filter*.
- IV. *Interference cancelling* [Fig. 1.7(d)]. In this final class of applications, the adaptive filter is used to cancel *unknown interference* contained (alongside an information-bearing signal component) in a *primary signal*, with the cancellation being optimized in some sense. The primary signal serves as the desired response for the adaptive filter. A *reference (auxiliary) signal* is employed as the input to the adaptive filter. The reference signal is derived from a sensor or set

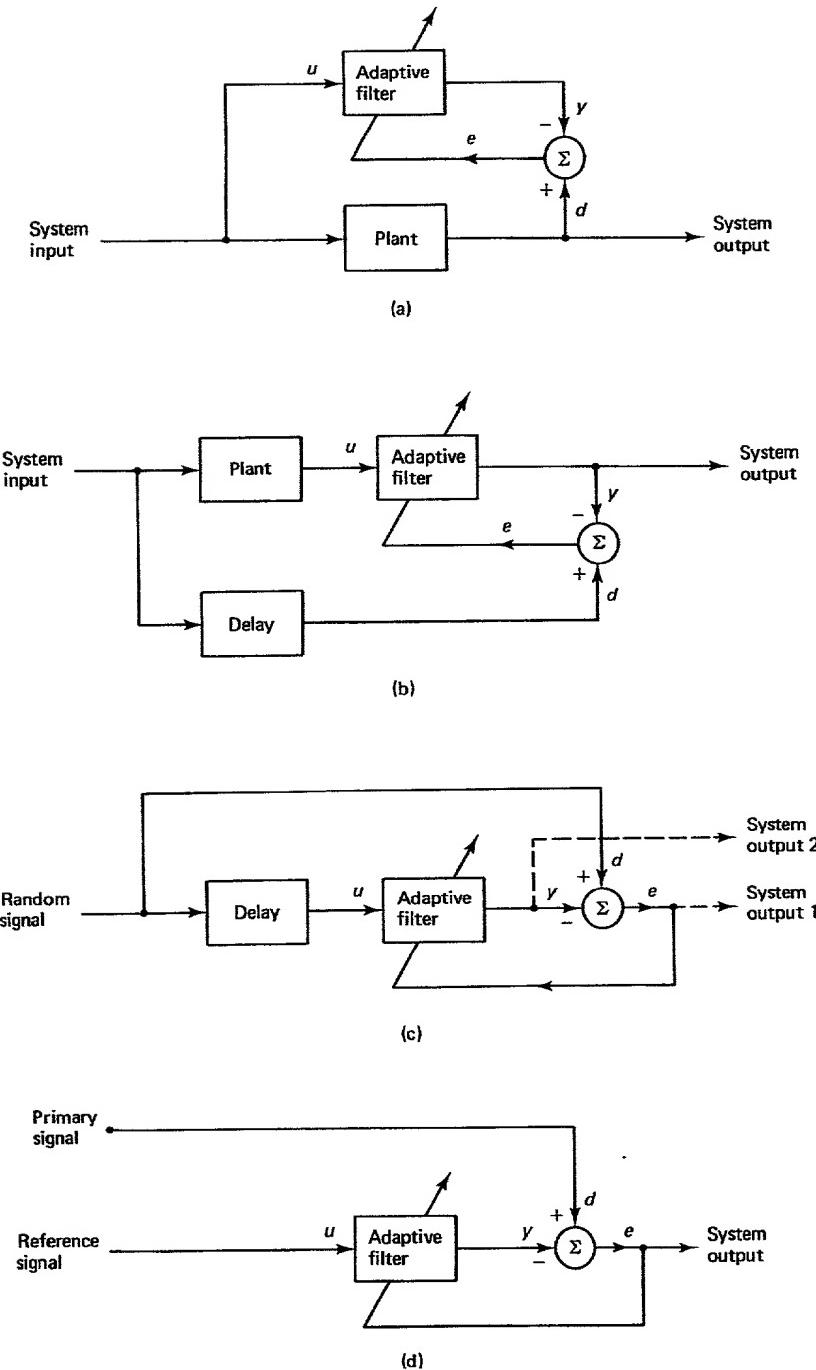


Figure 1.7 Four basic classes of adaptive filtering applications: (a) class I: identification; (b) class II: inverse modeling; (c) class III: prediction; (d) class IV: interference cancelling.

of sensors located in relation to the sensor(s) supplying the primary signal in such a way that the information-bearing signal component is weak or essentially undetectable.

In Table 1.1 we have listed some applications that are illustrative of the four basic classes of adaptive filtering applications. These applications, totaling twelve, are drawn from the fields of control systems, seismology, electrocardiography, communications, and radar.⁶ They are described individually in the remainder of this section.

TABLE 1.1 APPLICATIONS OF ADAPTIVE FILTERING

Class of adaptive filtering	Application
I. Identification	System identification Layered earth modeling
II. Inverse modeling	Predictive deconvolution Adaptive equalization
III. Prediction	Linear predictive coding Adaptive differential pulse-code modulation Autoregressive spectrum analysis Signal detection
IV. Interference cancelling	Adaptive noise cancelling Echo cancellation Radar polarimetry Adaptive beamforming

System Identification

System identification is the experimental approach to the modeling of a process or a plant (Åström and Wittenmark, 1990; Söderström and Stoica, 1988; Ljung, 1987; Ljung and Söderström, 1983; Goodwin and Payne, 1977). It involves the following steps: experimental planning, the selection of a model structure, parameter estimation, and model validation. The procedure of system identification, as pursued in practice, is iterative in nature in that we may have to go back and forth between these steps until a satisfactory model is built. Here we discuss briefly the idea of adaptive filtering algorithms for estimating the parameters of an unknown plant modeled as a transversal filter.

Suppose we have an unknown dynamic plant that is linear and time varying. The plant is characterized by a *real-valued* set of discrete-time measurements that describe the variation of the plant output in response to a known stationary input. The requirement is to develop an *on-line transversal filter model* for this plant, as illustrated in Fig. 1.8. The model consists of a finite number of unit-delay elements and a corresponding set of adjustable parameters (tap weights).

Let the available input signal at time n be denoted by the set of samples: $u(n)$, $u(n - 1)$, ..., $u(n - M + 1)$, where M is the number of parameters in the

⁶For additional applications of adaptive filtering, see Widrow and Stearns (1985), Cowan and Grant (1985), and the special issues on adaptive filters listed in the References and Bibliography at the end of the book.

From Eq. (14.80), the factor $\gamma(n)$ is the ratio of the a priori estimation error $\alpha(n)$ to the a posteriori estimation error $e(n)$; that is,

$$\gamma(n) = \frac{\alpha(n)}{e(n)} \quad (14.82)$$

We refer to $\gamma(n)$ as the *conversion factor*. The significance of it is that it converts the a priori estimation error $\alpha(n)$, based on the old tap-weight vector $\hat{w}(n-1)$, into the a posteriori estimation error $e(n)$, based on the updated value $w(n)$. Note that

$$0 \leq \gamma(n) \leq 1, \quad \text{for all } n \quad (14.83)$$

We will have a great deal more to say about the conversion factor $\gamma(n)$ in the next chapter.

In this section, we have presented a method for a *direct extraction* of the a posteriori estimation error $e(n)$ without having to compute the weight vector $\hat{w}(n)$. The presentation has been somewhat lengthy, the motivation being to provide interpretations of the elements that make up $e(n)$. Indeed, the main result of the section described in Eq. (14.80) may be derived in a much simpler fashion (Shepherd and McWhirter, 1991), the reader is referred to Problem 6 for details.

14.10 SYSTOLIC ARRAY IMPLEMENTATION II

Figure 14.7 shows the systolic array implementation of the *modified QRD-RLS algorithm* (McWhirter, 1983). Apart from the introduction of an extra parameter $\gamma^{1/2}$ into the boundary cell, the boundary and internal cells in the structure of Fig. 14.7 are identical to those shown in the triangular systolic array section of Fig. 14.2.

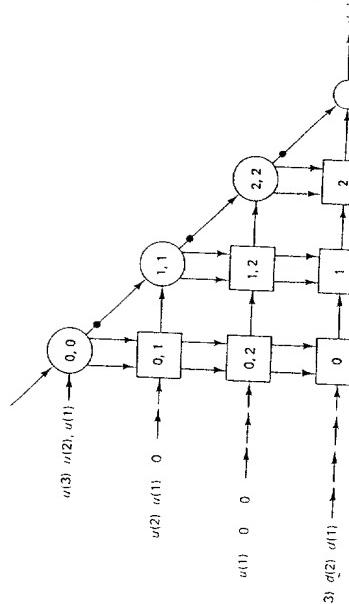


Figure 14.7 Systolic array implementation of the modified QRD-RLS algorithm along the diagonal of the array represent storage elements. The processing delay, which is a consequence of the temporal skew imposed on the input data, may be incorporated within the associated boundary cell

Moreover, the linear systolic array section has been omitted from the structure of Fig. 14.7, as the modified QRD-RLS algorithm eliminates the need for computing the weight vector $\hat{w}(n)$ explicitly. However, the structure of Fig. 14.7 includes a new element referred to as the *final processing cell* (indicated by a small circle). This cell is intended to compute the a posteriori estimation error $e(n)$. The specific functions of the boundary cells, the internal cells, and final processing cell are given in Fig. 14.8.

We remarked earlier that the scaling factor $\gamma^{1/2}(n)$, defined in Eq. (14.57) is generated as a result of the orthogonal triangulation process. This process manifests itself in Fig. 14.7 in the sense that the last boundary cell of the triangular section produces an output equal to $\gamma^{1/2}(n)$. Correspondingly, the last cell in the row of internal cells appended to the triangular section produces an output equal to

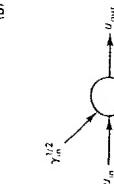
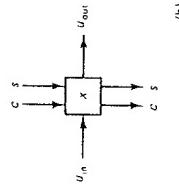
If $u_n = 0$ then

$c = 1$,
 $s = 0$, and
 $\gamma_{out}^{1/2} = \gamma_{in}^{1/2}$
 $x \leftarrow \gamma^{1/2}x$

Otherwise

$x \leftarrow -\sqrt{\lambda}x^2 - \gamma_{in}^{1/2}$
 $c = \frac{x}{\sqrt{\lambda}}$
 $s = \frac{u_n}{x}$
 $\gamma_{out}^{1/2} \leftarrow \gamma_{in}^{1/2}$

(a)



(b)

Initialization
At $n = 0$, set
 $x = 0$,
 $c = 1$,
 $s = 0$,
 $\gamma_{in} = 1$

$x \leftarrow \gamma^{1/2}x$

Otherwise

$x \leftarrow -\sqrt{\lambda}x^2 - \gamma_{in}^{1/2}$

$c = \frac{x}{\sqrt{\lambda}}$

$s = \frac{u_n}{x}$

$\gamma_{out}^{1/2} \leftarrow \gamma_{in}^{1/2}$

(c)

Figure 14.8 Cells for the modified QRD-RLS algorithm (a) boundary cell,
(b) internal cell, (c) final cell.

Note. The stored value x is initialized to be zero (i.e., real). For the boundary cell, it always remains real. Hence, the formulas for the rotation parameters c and s computed by the boundary cell can be simplified considerably, as shown in part (a). Note also that the values x stored in the array are elements of \mathbf{R}^* , here, $r^* = x$ for all elements of the array

$\gamma^{1/2}(n)\alpha(n)$. Thus, the final processing cell simply multiplies $\gamma^{1/2}(n)\alpha(n)$ thus produced by the output $\gamma^{1/2}(n)$ from the last boundary cell to produce the a posteriori estimation error $e(n)$.

As the time-skewed input data vectors enter the systolic array of Fig. 14.7, we find that updated estimation errors are produced at the output of the array at the rate of one every clock cycle. The estimation error produced on a given clock cycle corresponds, of course, to the particular element of the desired response vector $d(n)$ that entered the array M clock cycles previously.

It is noteworthy that the a priori estimation error $\alpha(n)$ may be obtained by dividing the output that emerges from the last cell in the appended (bottom) row of internal cells by the output from the last boundary cell. Also, the conversion factor $\gamma(n)$ may be obtained simply by squaring the output that emerges from the last boundary cell.

Figure 14.9 summarizes, in a diagrammatic fashion, the flow of signals in the systolic array of Fig. 14.7. The figure includes the external inputs $u(n)$ and $d(n)$, the resulting transformations in the internal states of the triangular section and appended row of internal cells, the respective outputs of these two sections, and the overall output of the complete processor. Note that the systolic processor I of Fig. 14.2 and the systolic processor II of Fig. 14.7 are mathematically equivalent (though not numerically).

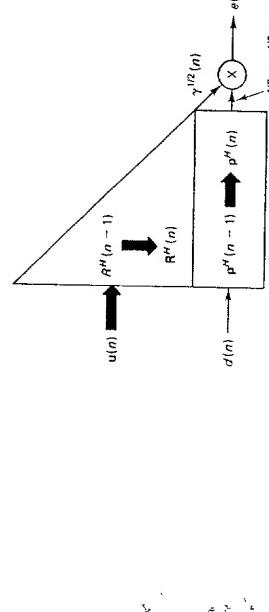


Figure 14.9. Diagrammatic representation of the flow of signals in the systolic array of Fig. 14.7

let $u(n)$ denote the input vector and $d(n)$ denote the desired response, both at time n . Given that the weight vector at this time is $\hat{w}(n)$, the corresponding a posteriori estimation error is

$$e(n) = d(n) - \hat{w}^T(n)u(n) \quad (14.84)$$

Suppose that the state of the array is frozen at time n_* , immediately after the systolic computation at time n is completed. Specifically, any update of stored values in the array is suppressed; otherwise, it is permitted to function normally in all other respects. At time n_* , we also set the desired response $d(n)$ equal to zero. We now define an input vector that consists of a string of zeros, except for the i th element that is set equal to unity, as shown by

$$\mathbf{u}'(n_*) = [0 \dots 0 \overset{\downarrow}{1} 0 \dots 0] \quad (14.85)$$

Thus, substituting these values in Eq. (14.84), we get

$$e(n_*) = -\hat{w}_i^*(n_*) \quad (14.86)$$

In other words, except for a trivial sign change, we may compute the i th element of the M -by-1 weight vector $w^*(n)$ by freezing the state of the processor at time n_* , and subsequently setting the desired response equal to zero, and feeding the processor with an input vector whose i th element is unity and the remaining $M - 1$ elements are all zero. The essence of all of this is that the complex-conjugated weight vector $\hat{w}^*(n)$ may be viewed as the impulse response of the nonadaptive (i.e., frozen) form of the systolic array processor in the sense that it can be generated as the system output produced by inputting an $(M - 1)$ -by- $(M - 1)$ identity matrix to the main triangular array and a zero vector to the bottom row of the array in Fig. 14.7 (Shepherd and McWhirter, 1991).

To "flush" the entire M -by-1 weight vector $\hat{w}^*(n)$ out of the systolic processor in Fig. 14.7, the procedure is therefore simply to halt the update of all stored values and input a data matrix that consists of a unit diagonal matrix (i.e., identity matrix) of dimension M . A demonstration of this procedure is given in the next section.

14.11 COMPUTER EXPERIMENT II: ADAPTIVE PREDICTION USING SYSTOLIC STRUCTURE II

Computation of the Weight Vector

A distinctive feature of the systolic structure shown in Fig. 14.7 is that unlike the systolic structure of Fig. 14.2, its operation does not involve the explicit computation of the weight vector $\hat{w}(n)$. Clearly, the structure of Fig. 14.7 may be extended to include a linear systolic section (as in Fig. 14.2) so as to compute $\hat{w}(n)$, if so required. However, there is a simpler method of computing the weight vector $\hat{w}(n)$ as a useful by-product of the direct error (residual) extraction capability inherent in the systolic process of Fig. 14.7.

The method for extracting the weight vector $\hat{w}(n)$ is referred to as *serial weight flushing* (Ward et al., 1986; Shepherd and McWhirter, 1991). To explain the method,

In this section, we revisit the computer experiment on adaptive prediction that we studied in Section 14.7, except that this time we use the single-section systolic processor of Fig. 14.7 for the computation. In particular, we are interested in studying the learning curve of the processor for different levels of ill conditioning of the input data. The learning curve is computed by plotting the ensemble-averaged squared value of the a priori estimation error $e(n)$ versus time n .

Using the autoregressive data base used for the computer experiment in Section 14.7, the learning curves shown in Fig. 14.10 are computed by ensemble averaging over 100 independent realizations of the systolic processor. The eigenvalue spreads used in the computation are 3, 10, and 100.

Predictor Coefficients

To compute the 2-by-1 weight vector defining the predictor coefficients, we may use the "weight flushing" procedure described in Section 14.10. Specifically, a computer realization of the systolic processor in Fig. 14.7 was performed; the learning process was halted after 25 iterations, and the state of the processor was subsequently frozen. By inputting the data matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

into the processor, the predictor coefficients $\hat{\mathbf{w}}(n)$ and $\hat{\mathbf{w}}(n)$ for $n = 25$ are flushed out. One hundred independent trials of this computation were performed, and the ensemble-averaged values of $\hat{\mathbf{w}}(25)$ and $\hat{\mathbf{w}}(25)$ were calculated for the eigenvalue spreads of 3, 10, and 100. The results so obtained are summarized in Table 14.6. This table also includes the corresponding optimum Wiener solution for the 2-by-1 weight vector as well as the result obtained by using the standard RLS algorithm for 25 iterations.

TABLE 14.6 OUTPUT OF SYSTOLIC ARRAY II AND COMPARISON WITH RLS ALGORITHM

$X(R)$	Experiment		Optimum		Relative error (%)
	$\hat{\mathbf{w}}_0$	$\hat{\mathbf{w}}_1$	w_0	w_1	
Systolic array II, $n = 25$					
3	0.9555	-0.91519	0.9750	-0.95	2.0
10	1.5233	-0.88600	1.5955	-0.95	4.5
100	1.8252	-0.87115	1.9114	-0.95	8.3

From this table, we can make the following observations:

1. After 25 iterations, the experimentally computed values of the predictor coefficients closely approach the corresponding optimum values.
2. Convergence of the systolic processor in Fig. 14.7 in the mean is essentially independent of the eigenvalue spread of the input data.

Convergion Factor

Figure 14.11 shows the time evolution of a single realization of the conversion $\gamma(n)$ for three different values of the eigenvalue spread: 3, 10, and 100. For each time n the computation of $\gamma(n)$ was performed simply by squaring the output that emerges from the last boundary cell of the systolic processor

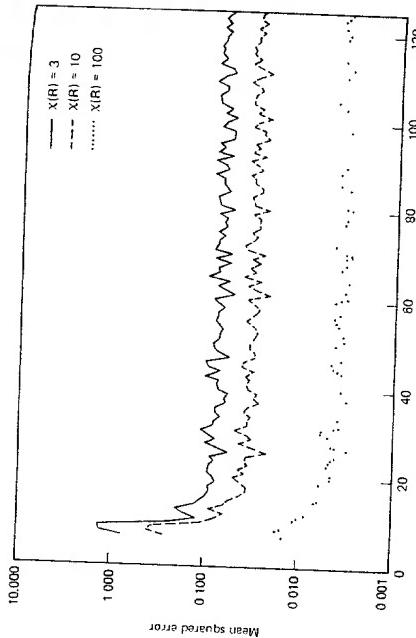


Figure 14.10 Learning curves of systolic array II for varying eigenvalue spreads

Table 14.5 summarizes estimates of the final mean-squared error $J(\infty)$ and the corresponding values of the misadjustment $M = (J(\infty)/J_{m,n}) - 1$ for these eigenvalue spreads. The estimates of $J(\infty)$ were each obtained by "time averaging" over the last 100 points of the pertinent learning curve.

TABLE 14.5 SUMMARY OF RESULTS FOR COMPUTER EXPERIMENT II

Eigenvalue spread, $X(R)$	Measured steady-state value of mean-squared error	Minimum mean-squared error	Misadjustment, $M(\%)$	RLS, $n = 25$
3	0.076568	0.0731	4.7	3
10	0.03632	0.0322	4.4	10
100	0.00980	0.0038	4.8	100

From the results presented in Fig. 14.10 and Table 14.5, we can draw the following conclusions.

1. The systolic processor of Fig. 14.7, used as an adaptive predictor, achieves good convergence in the mean square in about three or four iterations, this is about twice the number of elements in the coefficient vector.
2. The small values of the misadjustment M (all less than 5 percent) demonstrate that the systolic processor is capable of producing results close to the optimum Wiener solution.
3. The convergence properties of the systolic processor are relatively insensitive to variations in the eigenvalue spread of the input data.

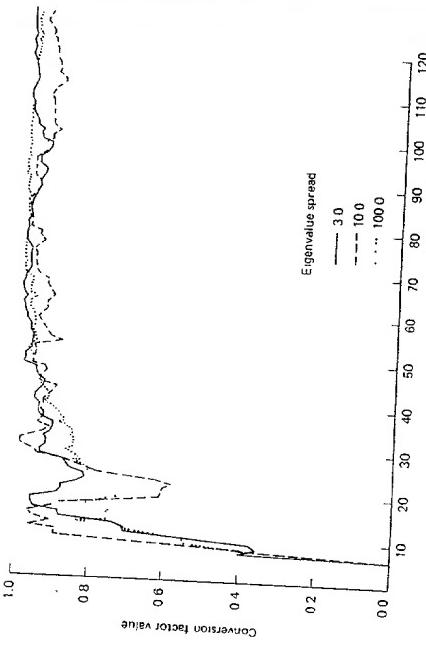


Figure 14.11 Single realizations of the conversion factor $\gamma(n)$ for varying eigenvalue spread

From Fig. 14.11, we may make the following observations

1. The initialization period, for which the conversion factor $\gamma(n)$ is zero, occupies six iterations, that is, $3M$.
2. After the completion of initialization, the conversion factor is essentially independent of the eigenvalue spread of the underlying correlation matrix of the input data.
3. The conversion factor converges rapidly toward the steady-state (final) value of unity. Specifically, it reaches a value of approximately 0.6 after about four or five iterations. This confirms our earlier observation that for the adaptive predictor of order 2 considered herein the systolic processor II converges in about four or five iterations, which is roughly equal to $2M$.

14.12 ADAPTIVE BEAMFORMING

An important practical application of adaptive filter theory is in *adaptive beamforming*. The need for this special form of adaptive spatial filtering arises in such diverse fields as radar, sonar, and communications. The basic objective of an adaptive beamformer is to identify a set of complex weights for modifying the outputs of an array of sensors so as to produce a far-field pattern that optimizes the reception of a target signal along a direction of interest in some statistical sense. The sensors may consist of antenna elements as in radar and communications, or hydrophones as in sonar. In any event, the adaptive processing may be performed in either one of two spaces:

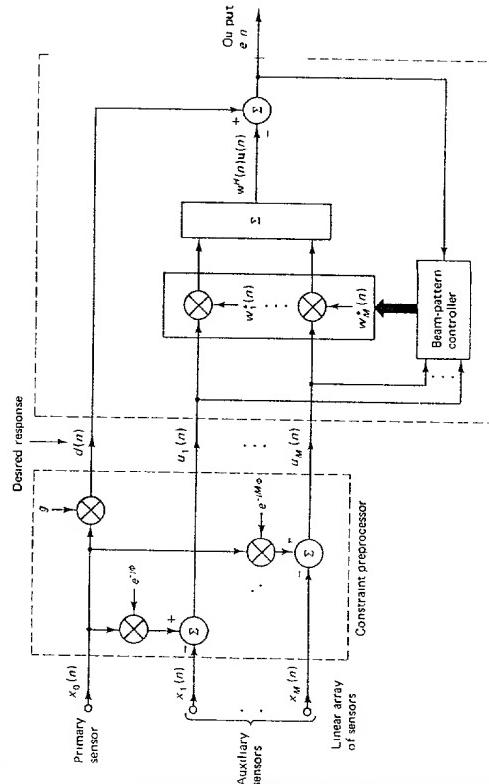


Figure 14.12 Structural layout of adaptive beamformer
Sec. 14.12 Adaptive Beamforming

sensor outputs measured simultaneously at time n , be written in the partitioned form:

$$\begin{aligned} \text{input vector} &= \begin{bmatrix} x_0(n) \\ \vdots \\ x_M(n) \end{bmatrix} \\ &= \begin{bmatrix} x_0(n) \\ \vdots \\ x(n) \end{bmatrix} \end{aligned} \quad (14.87)$$

where $x_0(n)$ is the output of sensor 0 and $x(n)$ is the vector of remaining M sensor outputs. Let this input vector be applied to a *linear combiner*, producing the output

$$\begin{aligned} e(n) &= [w_0^*(n), w^H(n)] \begin{bmatrix} x_0(n) \\ \vdots \\ x(n) \end{bmatrix} \\ &= w_0^*(n)x_0(n) + w^H(n)x(n) \end{aligned} \quad (14.88)$$

where the scalar $w_0(n)$ and the vector $w(n)$ constitute an $(M+1)$ -by-1 weight vector for the entire array. Suppose that we wish to track a target in the far field of the array that lies along a direction specified by the electrical angle ϕ . This electrical angle defines an $(M+1)$ -by-1 steering vector, which we may write as follows:

$$\begin{aligned} \text{steering vector} &= \begin{bmatrix} 1 \\ e^{-j\phi} \\ e^{-j2\phi} \\ \vdots \\ e^{-jM\phi} \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ \vdots \\ e(\phi) \end{bmatrix} \end{aligned} \quad (14.89)$$

where $e(\phi)$ is itself defined by

$$e'(\phi) = [e^{-j\phi}, e^{-j2\phi}, \dots, e^{-jM\phi}] \quad (14.90)$$

In order to provide "protection" for the target signal being tracked along the direction specified by the electrical angle ϕ , we impose a *linear beam constraint* of the following form:

$$[w_0^*(n), w^H(n)] \begin{bmatrix} 1 \\ e(\phi) \end{bmatrix} = g$$

or, equivalently,

$$w_0^*(n) + w^H(n)e(\phi) = g \quad (14.91)$$

where g is some constant. The constraint of Eq. (14.91) ensures that the beamformer provides a constant gain g along the direction specified by the electrical angle ϕ regardless of the values assigned to $w_0(n)$ and $w(n)$. In particular, if the sensors are

uniformly excited by a distant source (in the far field of the array) along a look direction specified by the electrical angle ϕ , and each sensor output has a magnitude equal to unity, we may then set the sensor outputs as follows:

$$x_0(n) = 1$$

and

$$x(n) = e(\phi)$$

Substituting these values in Eq. (14.88) and using the constraint described in Eq (14.91) yields an array output $e(n)$, and therefore a gain for the beamformer, equal to the constant g .

The overall operation of the beamformer is thus described by the pair of equations: (14.88) and (14.91). We wish to combine these two equations into a single relation that has the customary form

$$e(n) = d(n) - w^H(n)u(n) \quad (14.92)$$

where $u(n)$ is an M -by-1 input vector; $d(n)$ may be viewed as a "desired response" in which case $e(n)$ represents an error signal. Both $d(n)$ and $u(n)$ are to be defined. We may achieve this aim by using Eq. (14.91) to eliminate $w_0^*(n)$ from Eq (14.88). Specifically, we may rewrite the beamformer output $e(n)$ as follows:

$$\begin{aligned} e(n) &= \underbrace{g x_0(n)}_{\text{desired response}} - \underbrace{w^H(n)[x_0(n)e(\phi) - x(n)]}_{\text{input vector } u(n)} \end{aligned} \quad (14.93)$$

The implications of using Eq (14.93) to define the beamformer output are threefold

1. By eliminating the unknown weight w_0 from the beamformer output, the number of degrees of freedom available for designing the beamformer is reduced by one to M , the dimension of the weight vector $w(n)$.
2. The signal received from sensor 0 in the array, scaled by the constant gain g is the "desired response" $d(n)$.
3. The "input vector" $u(n)$ is completely determined by the specified electrical angle ϕ and the signals received from the entire array.

Consequently, the beamformer design may be optimized by minimizing a cost function based on the "error signal" $e(n)$. This, in turn, means that we may use an adaptive filtering algorithm of the type developed in the present and previous chapters of the book to compute the adjustable weight vector $w(n)$.

The formulation of the error signal $e(n)$ as in Eq. (14.93) suggests that we structure the adaptive beamformer as shown in Fig. 14.12 (Ward et al., 1986). According to this structure, the linearly constrained adaptive beamformer² consists of two key components:

² For the general least-squares (adaptive) beamformer with a multitude of linear constraints, see McWhirter and Shepherd (1986) and Stock and Kailath (1989).

1. **Constraint preprocessor.** This first component is shown on the left side of Fig. 14.12. It transforms the combination of the signals $x_0(n), x_1(n), \dots, x_k(n)$ received from the entire array of $M + 1$ sensors and the steering vector specified by the electrical angle ϕ linearly into two sets of signals: the desired response $d(n)$ and the input vector $u(n)$. According to the representation depicted in Fig. 14.12, in effect the first sensor (labeled 0) supplies the desired response $d(n)$; in the context of previous terminology, this element serves the role of a reference sensor. To fulfill such a function properly, the reference sensor is assumed to have an *omnidirectional* coverage (i.e., constant gain in all directions). The remaining M sensors of the array, responsible for the generation of the input vector $u(n)$, serve the role of *primary sensors*. Note that the constraint preprocessor ensures that any signal entering it from the specified look direction is removed from the input vector $u(n)$ supplied to the adaptive combiner. The adaptive combiner is thereby *prevented from nulling* the target signal that is being tracked along the look direction of interest.
2. **Adaptive combiner.** This second component is shown on the right side of Fig. 14.12. The specific details of the adaptive combiner are naturally determined by the type of adaptive filtering algorithm selected to compute the adjustable weight vector $w(n)$, given the input vector $u(n)$ and the desired response $d(n)$ from the constraint preprocessor.

In the context of the latter issue, and in the light of the material covered in this book, we may use any one of three adaptive filtering approaches:

1. **LMS approach.** This approach offers simplicity (and therefore cost-effectiveness) and numerical robustness. The major shortcoming of the LMS algorithm is its *poor convergence properties*, particularly when it operates in a signal environment of broad dynamic range. Thus, in turn, places a fundamental limitation on the use of the LMS algorithm in sophisticated electronic systems.
2. **Standard least-squares estimation.** Here we have two options available to us:
 - (a) We opt for *block estimation* to compute the least-squares estimate of the adjustable weight vector. That is, we use the system of normal equations to write [see Eq. (10.48)]

$$\hat{w} = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \mathbf{d} \quad (14.94)$$

where \mathbf{A} is the data matrix describing the time evolution of the input vector $u(n)$, and \mathbf{d} is the desired data vector describing the time evolution of the desired response $d(n)$, both on a snapshot-by-snapshot basis. In the context of adaptive beamforming, this approach is referred to in the literature as the *sample matrix inversion algorithm* (Reed et al., 1974).

- (b) We opt for a recursive approach and use the standard *recursive least-squares (RLS) algorithm* of Chapter 13.

By using the sample matrix inversion or the RLS algorithm, we overcome the problem of poor convergence associated with the LMS algorithm. However, both approaches have two major problems of their own. (1) *increased computational complexity* that cannot be easily overcome through the use of very large scale integration (VLSI), and (2) *numerical instability* resulting from the

use of finite-precision arithmetic and the requirement of inverting a large matrix.

3. **Data decomposition-based least-squares estimation.** This approach differs from approach 2 in that it involves *direct orthogonalization* of the data matrix. Basically, the orthogonalization may be rectangular or triangular in form, so we have two options available to us.
- (a) We use *rectangular* orthogonalization of the data matrix, which lends itself to block estimation. In particular, we apply the *singular-value decomposition* directly to the data matrix. Hence, we may express the solution to the linear least-squares problem as

$$\hat{w} = \mathbf{A}^+ \mathbf{d} \quad (14.95)$$

where \mathbf{A}^+ is the pseudoinverse of the data matrix \mathbf{A} , and \mathbf{d} is the desired data vector. For the computation of the singular-value decomposition, we use the *cyclic Jacobi* method, based on a sequence of plane (Givens) rotations. Details of this block estimation approach were presented in Chapter 11.

- (b) We use *triangular* orthogonalization of the data matrix, which lends itself to recursive estimation. In this case, the *QR-decomposition* is used to accomplish the direct orthogonalization of the data matrix. Here, we may use the conventional form of the QRD-RLS algorithm implemented⁴ using the two-section systolic array of Fig. 14.2, or alternatively, the modified form of the QRD-RLS algorithm implemented using the single-section systolic array of Fig. 14.7.

Both systolic approaches under 3(b) offer a fast rate of convergence that is inherent in least-squares estimation, and good numerical properties resulting from the widely accepted⁵ fact that the method of QR-decomposition by Givens rotations is one of the very best procedures for solving the linear least-squares problem. In practice, the modified systolic array II of Section 14.10 is preferred over the systolic array I of Section 14.6 because it offers a considerable reduction in the amount of the required computation and circuitry, since it avoids the use of a linear section that is an integral part of systolic array I. In particular, it is no longer necessary for us to clock out the elements of the lower triangular matrix $R^H(n)$ computed in the cells of the triangular section, solve for the weight vector $\hat{w}(n)$ by back substitution, or compute the adaptive beamformer output by forming the inner product $\hat{w}^H(n)u(n)$.

In summary, the combination of the constraint preprocessor and the modified systolic array processor II, described respectively in Figs. 14.3 and 14.7, provides an important method for solving a linearly constrained adaptive beamforming problem. In particular, this combination offers the following attractive features.

1. The constraint preprocessor steers the beamformer along a look direction of interest determined by the value of the electrical angle ϕ , and also protects a

⁴The recursive least-squares estimation for adaptive beamforming may also be solved in a numerically stable manner by using a sequence of Householder transformations (Rader and Stearns, 1986). However, the QRD-RLS algorithms described in this chapter have the advantage of offering a higher degree of parallelism; this is made possible by the use of the Givens rotations, which operate on $m \times (1 \text{ - } 2\text{-by-}2)$ blocks of data.

- target signal being tracked along that direction; the electrical angle ϕ is related to the actual incidence angle θ by Eq. (1.61).
2. The systolic array provides for the adaptive processing of the constraint preprocessor outputs so as to cancel interferences originating from unknown directions; it does so with a fast rate of convergence and in a numerically stable manner.

14.13 MINIMUM-VARIANCE DISTORTIONLESS BEAMFORMERS

In the preceding section, we showed that by means of a *constraint preprocessor*, it is possible to extend the use of systolic arrays I and II based on the QR-decomposition, to solve constrained least-squares estimation recursively as it would arise in adaptive beamforming applications. The order of signal processing in the constraint preprocessor method is depicted in Fig. 14.13(a). Here we note that the preprocessor consumes approximately M operations, while the least-squares processor uses approximately $\frac{1}{2}M^2$ operations, where M is the number of adjustable weights in the beamformer. If we wish to impose several (K , say) constraints independently (but not simultaneously), that is, if the same processing is to be done along K separate look directions, we then have to run the same data through the beamformer a total of K times. Such an imposition requires K different preprocessor configurations and K least-squares processing runs, taking a total of approximately $(KM + \frac{1}{2}KM^2)$ operations.

An alternative method is to use a *minimum-variance distortionless response (MVDR) beamformer*, in which the variance (i.e., average power) of the output is minimized under the constraint that a distortionless response is maintained along the

direction of a target signal of interest, hence the name of the beamformer. The MVDR beamformer is basically a *constraint postprocessor* in that it permits the application of the constraint *after* the least-squares processing, as depicted in Fig. 14.13(b). Consequently, the output from the least-squares processor is *broadcast* to K separate constraint processors, each of which uses approximately M operations (see below). The total number of operations in the MVDR beamformer is thus approximately $(\frac{1}{2}M^2 + KM)$; this is considerably less than the corresponding number of operations for the preprocessor-constraint method for large values of K and M . Accordingly, an MVDR beamformer has the advantage of being computationally more efficient than a preprocessor constraint-based beamformer, and also it may be performed potentially on the fly.

The MVDR Problem

Consider then a linear array of M uniformly spaced sensors whose outputs are individually weighted and then summed to produce the beamformer output

$$e(\ell) = \sum_{i=1}^M w_i^*(\ell) u_i(\ell) \quad (14.96)$$

where u_i is the output of sensor ℓ at time i , and $w_i(\ell)$ is the associated weight. To simplify the mathematical presentation, we consider the simple case of a single look direction. Let $s_1(\phi), s_2(\phi), \dots, s_M(\phi)$ be the elements of a prescribed *steering vector* $s(\phi)$, the electrical angle ϕ is determined by the look direction of interest. In particular, the element $s_\ell(\phi)$ is the output of sensor ℓ of the array under the condition that there is no signal other than that due to a source of interest. We may thus state the MVDR problem as follows:

Minimize the cost function

$$\mathcal{E}(n) = \sum_{i=1}^M \lambda^{n-i} |e(i)|^2 \quad (14.97)$$

subject to the constraint

$$\sum_{i=1}^M w_i^*(n) s_i(\phi) = 1, \quad \text{for all } n \quad (14.98)$$

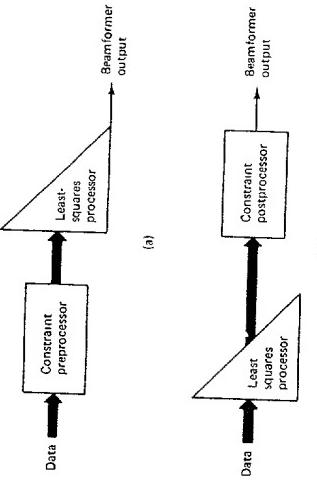
Using matrix rotation, we may redefine the cost function $\mathcal{E}(n)$ of Eq. (14.97) as

$$\mathcal{E}(n) = \mathbf{e}^H(n) \mathbf{\Lambda}(n) \mathbf{e}(n) \quad (14.99)$$

where $\mathbf{\Lambda}(n)$ is the exponential weighting matrix, and $\mathbf{e}(n)$ is the constrained beamformer output vector. Although Eq. (14.99) has the same mathematical form as that of Eq. (14.8), nevertheless, the vector $\mathbf{e}(n)$ has different interpretations in these two equations. According to Eq. (14.96), the beamformer output vector $\mathbf{e}(n)$ is related to the data matrix $\mathbf{A}(n)$ by

$$\begin{aligned} \mathbf{e}(n) &= [e(1), e(2), \dots, e(n)]^H \\ &= \mathbf{A}(n) \mathbf{w}(n) \end{aligned} \quad (14.100)$$

Figure 14.13 (a) Constraint preprocessor-based beamformer; (b) MVDR beamformer



Sec 14.13 Minimum-Variance Distortionless Response Beamformers

Recursive Least-Squares Systolic Arrays Chap. 14

where $\mathbf{w}(n)$ is the weight vector, and the data matrix $\mathbf{A}(n)$ is defined in terms of the snapshots $\mathbf{u}(1), \mathbf{u}(2), \dots, \mathbf{u}(n)$ by

$$\begin{aligned} \mathbf{A}^H(n) &= [\mathbf{u}(1), \mathbf{u}(2), \dots, \mathbf{u}(n)] \\ &= \begin{bmatrix} u_1(1) & u_1(2) & \cdots & u_1(n) \\ u_2(1) & u_2(2) & \cdots & u_2(n) \\ \vdots & \vdots & \ddots & \vdots \\ u_M(1) & u_M(2) & \cdots & u_M(n) \end{bmatrix} \quad (14.101) \end{aligned}$$

We may now restate the MVDR problem in matrix terms as follows:

Given the data matrix $\mathbf{A}(n)$ and the exponential weighting matrix $\Lambda(n)$, minimize the cost function

$$g(n) = \|\Lambda^{1/2}(n)\mathbf{A}(n)\mathbf{w}(n)\|^2$$

with respect to the weight vector $\mathbf{w}(n)$, subject to the constraint

$$\mathbf{w}^H(n)s(\phi) = 1, \quad \text{for all } n$$

where $s(\phi)$ is the steering vector for a prescribed electrical angle ϕ .

The solution to this constrained optimization problem is described by the MVDR formula (see Section 10.11)

$$\hat{\mathbf{w}}(n) = \frac{\Phi^{-1}(n)s(\phi)}{s^H(\phi)\Phi^{-1}(n)s(\phi)} \quad (14.102)$$

where $\Phi(n)$ is the M -by- M correlation matrix of the exponentially weighted sensor outputs averaged over n snapshots, and which is related to the data matrix $\mathbf{A}(n)$ as follows.

$$\Phi(n) = \mathbf{A}^H(n)\Lambda(n)\mathbf{A}(n) \quad (14.103)$$

In the remainder of this section we describe two adaptive systolic beamformers for solving the MVDR problem both of which are based on the QR-decomposition of the data matrix. The first method (due to Schreiber (1986)) builds on the Gentleman–Kung structure (i.e., systolic array processor I of Section 14.6). The second method (due to McWhirter and Shepherd (1989)) provides for a direct computation of the MVDR beamformer output by incorporating the McWhirter structure (i.e., systolic array processor II of Section 14.10). These two MVDR beamformers are considered in turn.

Systolic MVDR Beamformer I

From the discussion presented in Section 14.2, we recall that the correlation matrix $\Phi(n)$ equals the matrix product $\mathbf{R}^H(n)\mathbf{R}(n)$, where $\mathbf{R}(n)$ is the upper triangular matrix that results from the application of the QR-decomposition to the exponentially weighted data matrix $\mathbf{A}(n)$. We may therefore transform the solution given in Eq. (14.102) as follows.

$$\hat{\mathbf{w}} = \frac{\mathbf{R}^{-1}(n)\mathbf{R}^{-H}(n)s(\phi)}{s^H(\phi)\mathbf{R}^{-1}(n)\mathbf{R}^{-H}(n)s(\phi)} \quad (14.104)$$

where $\mathbf{R}^{-H}(n)$ is the Hermitian transpose of the inverse matrix $\mathbf{R}^{-1}(n)$. Define the auxiliary vector

$$\mathbf{a}(n) = \mathbf{R}^{-H}(n)s(\phi) \quad (14.105)$$

We then note that the denominator of Eq. (14.104) is a real-valued scalar equal to the inner product of the auxiliary vector $\mathbf{a}(n)$ with itself. As for the numerator, it is equal to the vector $\mathbf{a}(n)$ premultiplied by the inverse matrix $\mathbf{R}^{-1}(n)$. We may thus simplify Eq. (14.104) as

$$\hat{\mathbf{w}}(n) = \frac{\mathbf{R}^{-1}(n)\mathbf{a}(n)}{\mathbf{a}^H(n)\mathbf{a}(n)} \quad (14.106)$$

For the computation of the weight vector $\hat{\mathbf{w}}(n)$, we may proceed as follows (Schreiber, 1986):

1. *Updating of the upper triangular matrix.* The upper triangular matrix $\mathbf{R}(n-1)$ is obtained from the QR-decomposition of the data matrix $\mathbf{A}(n-1)$. At time n , a new data snapshot $\mathbf{u}(n)$ is received, and the data matrix is updated as $\mathbf{A}(n)$ in accordance with Eq. (14.36). Given $\mathbf{R}(n-1)$ and $\mathbf{u}(n)$, the updated value $\mathbf{R}(n)$ of the upper triangular matrix is computed using Eq. (14.44) reproduced here for convenience of presentation:

$$\begin{bmatrix} \mathbf{R}(n) \\ \vdots \\ \mathbf{O} \end{bmatrix} = \mathbf{T}(n) \begin{bmatrix} \lambda^{1/2}\mathbf{R}(n-1) \\ \vdots \\ \mathbf{O} \\ \mathbf{u}^H(n) \end{bmatrix} \quad (14.107)$$

where $\mathbf{T}(n)$ comprises a sequence of standard Givens rotations. This equation is implemented using a triangular systolic array similar to that described in Section 14.7.

2. *Updating of the auxiliary vector.* The updated value $\mathbf{a}(n)$ of the auxiliary vector is computed from the old value $\mathbf{a}(n-1)$ using the same sequence of Givens rotations that constitutes the transformation $\mathbf{T}(n)$. We may derive this recursion in two steps:

- (a) Consider the identity

$$\mathbf{I} = [\lambda^{1/2}\mathbf{R}^H(n-1), \mathbf{O}', \mathbf{u}(n)\mathbf{T}^H(n)\mathbf{T}(n) \begin{bmatrix} \lambda^{-1/2}\mathbf{R}^{-H}(n-1) \\ \mathbf{O} \\ \mathbf{0}' \end{bmatrix}] \quad (14.108)$$

where it is noted $\mathbf{T}(n)$ is a unitary matrix and therefore $\mathbf{T}^H(n)\mathbf{T}(n)$ equals an n -by- n identity matrix. The identity matrix \mathbf{I} on the left side of Eq. (14.108) has dimension M . Applying the Hermitian transposition to both sides of Eq. (14.44), we have

$$[\mathbf{R}^H(n), \mathbf{O}'] = [\lambda^{1/2}\mathbf{R}^H(n-1), \mathbf{O}', \mathbf{u}(n)\mathbf{T}^H(n)] \quad (14.109)$$

Therefore, the use of Eq. (14.109) in (14.108) yields

$$\mathbf{I} = [\mathbf{R}^H(n), \mathbf{O}^T \mathbf{T}(n) \begin{bmatrix} \lambda^{-1/2} \mathbf{R}^{-H}(n-1) \\ \vdots \\ \mathbf{O} \end{bmatrix}] \quad (14.110)$$

Equivalently, we may invoke the special structure of the transformation $\mathbf{T}(n)$ to write

$$\begin{bmatrix} \mathbf{R}^{-H}(n) \\ \vdots \\ \mathbf{x} \end{bmatrix} = \mathbf{T}(n) \begin{bmatrix} \lambda^{-1/2} \mathbf{R}^{-H}(n-1) \\ \vdots \\ \mathbf{O} \end{bmatrix} \quad (14.111)$$

where \mathbf{X} denotes a matrix of no specific interest. Equation (14.111) shows that the inverse Hermitian matrix $\mathbf{R}^{-H}(n-1)$ may be updated by applying exactly the same sequence of Givens rotations as that used to update the upper triangular matrix $\mathbf{R}(n-1)$.

- (b) Postmultiplying both sides of Eq. (14.111) by the steering vector $\mathbf{s}(\phi)$, and using the definition of the auxiliary vector $\mathbf{a}(n)$ given in Eq. (14.105), we get

$$\begin{bmatrix} \mathbf{a}(n) \\ \mathbf{x} \end{bmatrix} = \mathbf{T}(n) \begin{bmatrix} \lambda^{-1/2} \mathbf{a}(n-1) \\ \mathbf{O} \end{bmatrix} \quad (14.112)$$

where \mathbf{x} is a vector of no specific interest. Equation (14.112) is the desired recursion for updating the old value $\mathbf{a}(n-1)$ of the auxiliary vector. Indeed, this equation shows that using the same set of Givens rotations that constitute the transformation $\mathbf{T}(n)$, we may compute $\mathbf{a}(n)$ from $\mathbf{a}(n-1)$. Note that the update of $\mathbf{R}^{-H}(n)$ given in Eq. (14.111) implies the update of \mathbf{a} in Eq. (14.112), and vice versa.

3. *Computation of the weight vector.* Having computed the upper triangular matrix $\mathbf{R}(n)$ and the auxiliary vector $\mathbf{a}(n)$, we next proceed to compute the weight vector $\hat{\mathbf{w}}(n)$. Specifically, we use Eq. (14.104) to write

$$\mathbf{R}(n)\hat{\mathbf{w}}(n) = \rho(n)\mathbf{a}(n) \quad (14.113)$$

where $\rho(n)$ is a scalar defined by

$$\rho(n) = \frac{1}{\mathbf{a}^H(n)\mathbf{a}(n)} \quad (14.114)$$

Equation (14.113) has the same basic mathematical form as that of Eq. (14.21). We may therefore use the method of back substitution to solve Eq. (14.113) for the weight vector $\hat{\mathbf{w}}(n)$ at time n .

4. Time is incremented to $n+1$, and we go back to step 1 and the process is repeated.

It is important to note that for the MVDR beamforming algorithm described herein to work, the auxiliary vector $\mathbf{a}(n)$ must be *initialized* at some time (n_0 , say).

TABLE 14.7 SUMMARY OF MVDR BEAMFORMING ALGORITHM

<i>Initialization:</i> Initialize the auxiliary vector $\mathbf{a}(n)$ at some time (n_0 , say) for which the inverse matrix $\mathbf{R}^{-1}(n_0)$ exists.
<i>Computation</i>
1. Compute the upper triangular matrix $\mathbf{R}(n)$ from its old value $\mathbf{R}(n-1)$ using the recursion
$\begin{bmatrix} \mathbf{R}(n) \\ \vdots \\ \mathbf{O} \end{bmatrix} = \mathbf{T}(n) \begin{bmatrix} \lambda^{-1/2} \mathbf{R}(n-1) \\ \vdots \\ \mathbf{O} \end{bmatrix} \quad (14.115)$
where $\mathbf{U}(n)$ is the new data snapshot (vector) at time n , and $\mathbf{T}(n)$ is an n -by- n transform on defined by a sequence of standard Givens rotations
2. Using the same set of Givens rotations, compute the auxiliary vector $\mathbf{a}(n)$ from its old value $\mathbf{a}(n-1)$
$\begin{bmatrix} \mathbf{a}(n) \\ \vdots \\ \mathbf{x} \end{bmatrix} = \mathbf{T}(n) \begin{bmatrix} \lambda^{-1/2} \mathbf{a}(n-1) \\ \vdots \\ \mathbf{0} \end{bmatrix} \quad (14.116)$
where \mathbf{x} is a vector of no specific interest
3. At time n , use $\mathbf{R}(n)$ and $\mathbf{a}(n)$ to back-solve for $\mathbf{R}^{-1}(n)\mathbf{a}(n)$. Hence, compute the weight vector
$\hat{\mathbf{w}}(n) = \frac{\mathbf{R}^{-1}(n)\mathbf{a}(n)}{\mathbf{a}^H(n)\mathbf{a}(n)} \quad (14.117)$
4. Increment time to $n+1$, go back to step 1, and repeat the process

The reason for this initialization is that at time $n=0$, the value

$$\mathbf{a}(0) = \mathbf{R}^{-H}(0)\mathbf{s}(\phi)$$

does not exist if $\mathbf{R}(0) = \mathbf{0}$. It is at the stage specified by time n_0 that the inverse matrix $\mathbf{R}^{-1}(n_0)$ exists, which therefore permits the explicit computation of the init a value $\mathbf{a}(n_0)$.

Table 14.7 presents a summary of the MVDR beamforming algorithm I (Schreiber, 1986). This algorithm is computationally efficient: in the general case of K independent look directions it only requires approximately $\frac{1}{2}M^2 + KM$ operations per cycle time, which corresponds to the minimum computational complexity. Moreover, the algorithm is known to have excellent numerical properties (Schreiber, 1986). It may also be implemented efficiently on some form of a systolic array. However, it appears to be difficult to implement the complete algorithm in a fully pipelined fashion. For example, updating the upper triangular matrix $\mathbf{R}(n)$ requires top-to-bottom processing, whereas the method of back substitution requires bottom-to-top processing. A method for overcoming this difficulty is described below.⁵

⁵ Bojanczyk and Luk (1987) describe another method for a fully pipelined systolic computation of MVDR. The Bojanczyk-Luk MVDR processor, however, is more complicated than that due to McWhirter and Shepherd (1989). The description of the fully pipelined MVDR beamformer presented in the last part of this section is based on the paper by McWhirter and Shepherd

Systolic MVDR Beamformer II

Returning to Eq. (14.106), we may define the a posteriori estimation error at time n as

$$\begin{aligned} e(n) &= \hat{\mathbf{w}}^H(n)\mathbf{u}(n) \\ (14.115) \quad &= \frac{e'(n)}{\|\mathbf{a}(n)\|^2} \end{aligned}$$

where $e'(n)$ is another estimation error, defined by

$$e'(n) = \mathbf{a}'(n)\mathbf{R}^{-H}(n)\mathbf{u}(n) \quad (14.116)$$

Moreover, we may use Eq. (14.105) to write the steering vector $\mathbf{s}(\phi)$ in terms of the value of the auxiliary vector at time $n - 1$ as follows:

$$\mathbf{s}(\phi) = \mathbf{R}^H(n - 1)\mathbf{a}(n - 1) \quad (14.117)$$

Clearly, we may also express $\mathbf{a}(n - 1)$ in the equivalent form

$$\begin{aligned} \mathbf{s}(\phi) &= \lambda^{-1}[\lambda^{1/2}\mathbf{R}^H(n - 1) : \mathbf{O}^T : \mathbf{u}(n)] \begin{bmatrix} \lambda^{1/2}\mathbf{a}(n - 1) \\ \vdots \\ \lambda^{1/2}\mathbf{v}(n - 1) \\ \vdots \\ \mathbf{0} \end{bmatrix} \quad (14.118) \end{aligned}$$

where $\mathbf{v}(n - 1)$ is an $(n - M - 1)$ -by-1 arbitrary vector. Since $\mathbf{T}(n)$ is a unitary matrix, we have

$$\mathbf{T}^H(n)\mathbf{T}(n) = \mathbf{I}$$

Accordingly, we may inject this relation at the midpoint of the matrix product on the right side of Eq. (14.118) and so rewrite it in the equivalent form

$$\begin{aligned} \mathbf{s}(\phi) &= \lambda^{-1}[\lambda^{1/2}\mathbf{R}^H(n - 1) : \mathbf{O}^T : \mathbf{u}(n)]\mathbf{T}^H(n)\mathbf{T}(n) \begin{bmatrix} \lambda^{1/2}\mathbf{a}(n - 1) \\ \vdots \\ \lambda^{1/2}\mathbf{v}(n - 1) \\ \vdots \\ \mathbf{0} \end{bmatrix} \quad (14.119) \end{aligned}$$

Accordingly, substituting the Hermitian transposed version of Eq. (14.107) yields

$$\begin{aligned} \mathbf{s}(\phi) &= \lambda^{-1}[\mathbf{R}^H(n) : \mathbf{O}^T]\mathbf{T}(n) \begin{bmatrix} \lambda^{1/2}\mathbf{a}(n - 1) \\ \lambda^{1/2}\mathbf{v}(n - 1) \\ \mathbf{0} \end{bmatrix} \quad (14.120) \end{aligned}$$

We now recognize from Section 14.8 that the structure of the transformation $\mathbf{T}(n)$ is such that we may write

$$\begin{aligned} \mathbf{T}(n) \begin{bmatrix} \lambda^{1/2}\mathbf{a}(n - 1) \\ \lambda^{1/2}\mathbf{v}(n - 1) \\ \mathbf{0} \end{bmatrix} &= \begin{bmatrix} \pi(n) \\ \lambda^{1/2}\mathbf{v}(n - 1) \\ \epsilon^*(n) \end{bmatrix} \\ (14.121) \quad &= \begin{bmatrix} \pi(n) \\ \mathbf{v}(n) \end{bmatrix} \end{aligned}$$

Hence, using Eq. (14.121) in (14.120), we obtain

$$\begin{aligned} \mathbf{s}(\phi) &= \lambda^{-1}[\mathbf{R}^H(n) : \mathbf{O}^T] \begin{bmatrix} \pi(n) \\ \mathbf{v}(n) \end{bmatrix} \\ (14.122) \quad &= \lambda^{-1}\mathbf{R}^H(n)\pi(n) \end{aligned}$$

However, from Eq. (14.105) we also have

$$\mathbf{s}(\phi) = \mathbf{R}^H(n)\mathbf{a}(n)$$

It follows therefore that the vector $\pi(n)$ is simply a scaled version of the auxiliary vector $\mathbf{a}(n)$, as shown by

$$\pi(n) = \lambda\mathbf{a}(n) \quad (14.123)$$

We may thus view Eq. (14.121) as a straightforward recursion for computing the updated auxiliary vector $\mathbf{a}(n)$.

Table 14.8 presents a summary of MVDR beamforming algorithm II (McWhirter and Shepherd, 1989). The main point of difference between this MVDR algorithm and that summarized in Table 14.7 is that in the former we avoid the need for computing the weight vector $\hat{\mathbf{w}}(n)$. By so doing, a fully pipelined systolic implementation of the MVDR beamforming algorithm is feasible. We may develop the

TABLE 14.8 SUMMARY OF MVDR BEAMFORMING ALGORITHM II

Initialization: Initialize the auxiliary vector $\mathbf{a}(n)$ at some time n_0 for which the inverse matrix $\mathbf{R}^{-1}(n_0)$ exists.

Computation:

1. Update the upper triangular matrix $\mathbf{R}(n - 1)$ using the recursion

$$\begin{bmatrix} \mathbf{R}(n) \\ \vdots \\ \mathbf{0} \end{bmatrix} = \mathbf{T}(n) \begin{bmatrix} \lambda^{1/2}\mathbf{R}(n - 1) \\ \vdots \\ \mathbf{0} \end{bmatrix}$$

where $\mathbf{u}(n)$ is the new data snapshot at time n , and $\mathbf{T}(n)$ is an n -by- n transformation defined by a sequence of Givens rotations

2. Update the auxiliary vector $\mathbf{a}(n - 1)$ using the recursion

$$\begin{bmatrix} \lambda\mathbf{a}(n) \\ \vdots \\ \mathbf{0} \end{bmatrix} = \mathbf{T}(n) \begin{bmatrix} \lambda^{1/2}\mathbf{a}(n - 1) \\ \vdots \\ \mathbf{0} \end{bmatrix}$$

where $\mathbf{v}(n)$ is a vector of no specific interest, except for the final element that may be used to compute $e'(n)$

3. Given the updated values $\mathbf{R}(n)$ and $\mathbf{a}(n)$, compute the MVDR beamformer output at time n as

$$e(n) = \frac{e'(n)}{\|\mathbf{a}(n)\|}$$

where $e'(n)$ is the last element of the vector $\mathbf{v}(n)$, it may also be computed using

$$e'(n) = \mathbf{x}^H(n)\mathbf{R}^{-H}(n)\mathbf{u}(n)$$

4. Increment time n by 1, go back to step 1 and repeat the process

idea for a fully pipelined systolic implementation of the MVDR beamforming algorithm II by comparing the solution to the MVDR problem with the solution to the ordinary least-squares estimation problem. This comparison is displayed in Table 14.9. In particular, the pair of relations on the left side of this table is a reproduction of Eqs. (14.116) and (14.121) with $\lambda a(n)$ written in place of $\mathbf{m}(n)$. The first relation on the right side of the table follows from the definition of the a posteriori estimation error $e'(n)$ given in Eqs. (14.84) and (14.21), and the second relation is a reproduction of Eq. (14.48).

TABLE 14.9

Minimum-variance distortionless response	Least-squares estimation
$e'(n) = \mathbf{a}^H(n) \mathbf{R}^{-H}(n) \mathbf{u}(n)$	$e(n) = d(n) - \mathbf{p}^H(n) \mathbf{R}^{-H}(n) \mathbf{u}(n)$
$\begin{bmatrix} \lambda^{1/2} \mathbf{a}(n-1) \\ \mathbf{T}(n) \end{bmatrix} = \begin{bmatrix} \lambda \mathbf{a}(n) \\ \lambda^{1/2} \mathbf{v}(n-1) \\ \mathbf{0} \end{bmatrix}$	$\begin{bmatrix} \lambda^{1/2} \mathbf{p}(n-1) \\ \lambda^{1/2} \mathbf{v}(n-1) \\ d^*(n) \end{bmatrix} = \begin{bmatrix} \mathbf{p}(n) \\ \gamma^{1/2} n \alpha^*(n) \\ \text{where } \alpha(n) \text{ is the a priori estimation error} \end{bmatrix}$

Next referring to the systolic array of Fig. 14.7, we note that at time $n = 1$, the row vector $\mathbf{p}'(n-1)$ is generated by the bottom row of internal cells. The estimation error $e(n)$ is obtained from the output of the final cell at time n . Accordingly, if we replace the vector $\mathbf{p}'(n-1)$ by $\mathbf{a}'(n-1)$, and set the value of $d(n)$ in Fig. 14.7 equal to zero, then at time n we may make the following observations:

1. The output from the final cell in Fig. 14.7 is identically equal to $-\lambda e'(n)$.
2. The vector $\lambda \mathbf{a}''(n)$ is generated and stored in the bottom row of cells in Fig. 14.7.
3. The process may be continued in a simple recursive manner so as to generate the sequence of errors $e'(n)$, $e'(n+1)$,

These observations may be exploited to develop a fully pipelined systolic implementation of the MVDR beamforming algorithm II; for details see McWhirter and Shepherd (1989).

14.14 COMPUTER EXPERIMENT ON ADAPTIVE BEAMFORMING

In this section we illustrate the performance of the systolic array implementation of an adaptive MVDR beamformer by considering a linear array of five uniformly spaced sensors. The spacing d between adjacent elements equals one half of the received wavelength. The array operates in an environment that consists of a target signal and a single source of interference, which are noncoherent with each other. The exponential weighting factor $\lambda = 1$.

The aims of the experiment are two-fold:

The first aim is to compare the performance of the systolic array with that of a

1. To examine the evolution of the adapted spatial response (pattern) of the beamformer with time
2. To evaluate the effect of varying the interference-to-target ratio on the interference-nulling capability of the beamformer

The directions of the target and source of interference are as follows.

Excitation	Angle of incidence, θ , measured with respect to normal to the array (radians)
Target	$\sin^{-1}(0.2)$
Interference	$-\sin^{-1}(0.4)$ or 0

That is, the interference may originate from any one of two possible directions. The steering vector is defined by

$$\mathbf{s}'(\phi) = [1, e^{-j\phi}, e^{-j2\phi}, e^{-j3\phi}, e^{-j4\phi}]$$

where $\phi = \pi \sin \theta$

Figure 14.14 shows the individual effects of the parameters of interest on the adapted response of the beamformer. The initialization period occupies a total of 10 snapshots. The response is obtained by plotting $20 \log_{10} |\mathbf{w}^H(n) \mathbf{s}(\phi)|$ versus $\sin \theta = \phi/\pi$. The weight vector $\mathbf{w}(n)$ is computed by using the theory described in Section 14.1.3. The target-to-noise ratio is held constant at 10 dB. In the first part of Fig. 14.14(a), pertaining to a data base of 20 snapshots after the initialization has been completed, the three curves correspond to interference-to-noise ratios = 40, 30, 20 dB. In the second part, the data base is increased to 100 snapshots, and in the last part it is increased to 200 snapshots, again after the initialization has been completed. Part (a) of Fig. 14.14 corresponds to an angle of arrival $\theta = -\sin^{-1}(0.4)$ radians for the interference. Part (b) of the figure corresponds to $\theta = 0$ for the interference.

Based on these results, we may make the following observations:

1. The response of the beamformer is held fixed at a value of one under all conditions, as required.
2. With as few as 30 snapshots, including initialization, the beamformer exhibits a reasonably effective nulling capability, which continually improves as the beamformer processes more snapshots.
3. The response of the beamformer is relatively insensitive to variations in the interference-to-target ratio

14.15 EXTENSION OF SYSTOLIC ARRAY II FOR PARALLEL WEIGHT EXTRACTION

The systolic array processors considered in previous sections of the chapter have been somewhat limited in scope in the sense that they provide for a direct computation of the weight vector $\mathbf{w}(n)$ or the a posteriori estimation error $e'(n)$, but not both

simultaneously. In this section we describe an extension of the systolic array processor II that is indeed capable of directly computing the a posteriori estimation error $e(n)$ and simultaneously extracting least-squares weight vector $\hat{\mathbf{w}}(n)$ (Shepherd et al 1988; McWhirter, 1989).

To initiate the development of such a technique, we define an *augmented* data matrix $\mathbf{A}'(n)$ that consists of the data matrix $\mathbf{A}(n)$ extended by the desired data vector $\mathbf{d}(n)$ in the following fashion.

$$\mathbf{A}'(n) = [\mathbf{A}(n), \mathbf{d}(n)] \quad (14.124)$$

Thus, given that $\mathbf{A}(n)$ is an n -by- M matrix, $\mathbf{d}(n)$ is an n -by-1 vector, it follows that $\mathbf{A}'(n)$ is an n -by- $(M+1)$ matrix. Following the notation introduced in Section 14.2, we define the QR-decomposition of $\mathbf{A}'(n)$ as [see Eq. (14.14)]

$$\mathbf{Q}(n)\mathbf{A}'^{1/2}(n)\mathbf{A}'(n) = \begin{bmatrix} \mathbf{R}(n) \\ \mathbf{O} \end{bmatrix}$$

Correspondingly, we use Eq. (14.17) to write

$$\mathbf{Q}(n)\mathbf{A}'^{1/2}(n)\mathbf{d}(n) = \begin{bmatrix} \mathbf{p}(n) \\ \mathbf{v}(n) \end{bmatrix} \quad (14.125)$$

Hence, premultiplying both sides of Eq. (14.124) by $\mathbf{Q}(n)\mathbf{A}'^{1/2}(n)$ yields

$$\mathbf{Q}(n)\mathbf{A}'^{1/2}(n)\mathbf{A}'(n) = \begin{bmatrix} \mathbf{R}(n) & \mathbf{p}(n) \\ \mathbf{O} & \mathbf{v}(n) \end{bmatrix} \quad (14.126)$$

To complete the QR-decomposition of the augmented data matrix $\mathbf{A}'(n)$, we apply a sequence of Givens rotations that are designed to annihilate all the elements of $\mathbf{v}(n)$ except for its first element. The result of this operation is denoted by

$$\mathbf{Q}'(n)\mathbf{A}'^{1/2}(n)\mathbf{A}'(n) = \begin{bmatrix} \mathbf{R}(n) & \mathbf{p}(n) \\ \mathbf{O}^T & r(n) \\ \mathbf{O} & \end{bmatrix} \quad (14.127)$$

where $r(n)$ is some real-valued number; $r(n)$ has to be real because it is a diagonal element. Let $\mathbf{R}'(n)$ denote the $(M+1)$ -by- $(M+1)$ upper triangular matrix resulting from the QR-decomposition of the augmented data matrix $\mathbf{A}'(n)$. That is,

$$\mathbf{R}'(n) = \begin{bmatrix} \mathbf{R}(n) & \mathbf{p}(n) \\ \mathbf{O}^T & r(n) \end{bmatrix} \quad (14.128)$$

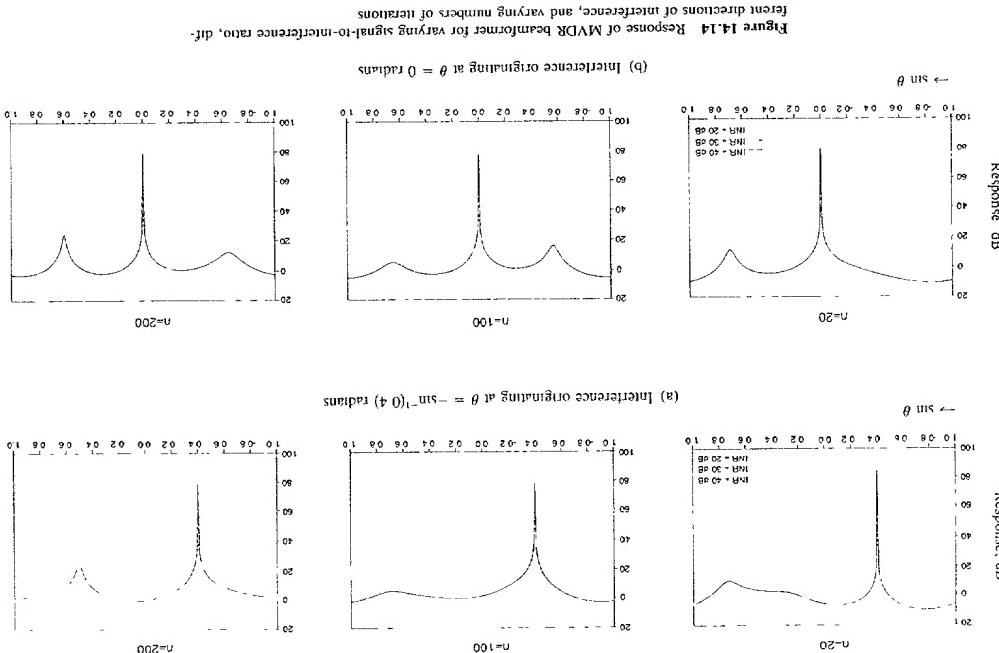
Next, we define an extended weight vector $\mathbf{w}'(n)$ of dimension $M+1$, related to the M -by-1 least-squares weight vector $\hat{\mathbf{w}}(n)$ as follows:

$$\mathbf{w}'(n) = \begin{bmatrix} -\hat{\mathbf{w}}(n) \\ 1 \end{bmatrix} \quad (14.129)$$

Hence, premultiplying $\mathbf{w}'(n)$ by $\mathbf{R}'(n)$, we get

$$\begin{aligned} \mathbf{R}'(n)\mathbf{w}'(n) &= \begin{bmatrix} \mathbf{R}(n) & \mathbf{p}(n) \\ \mathbf{O}^T & r(n) \end{bmatrix} \begin{bmatrix} -\hat{\mathbf{w}}(n) \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} -\mathbf{R}(n)\hat{\mathbf{w}}(n) + \mathbf{p}(n) \\ r(n) \end{bmatrix} \end{aligned} \quad (14.129)$$

Sec. 14.15 Extension of Systolic Array II for Parallel Weight Extraction



But from Eq. (14.21) we have

$$\mathbf{R}(n)\hat{\mathbf{w}}(n) = \mathbf{p}(n), \quad n > M$$

We may, therefore simplify Eq. (14.129) for $n > M$ as

$$\begin{aligned} \mathbf{R}(n)\mathbf{w}'(n) &= \begin{bmatrix} 0 \\ r(n) \end{bmatrix} \\ &= r(n) \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{aligned} \quad (14.130)$$

In other words, the matrix product $\mathbf{R}'(n)\mathbf{w}'(n)$ consists simply of the scalar $r'(n)$ multiplied by a punning vector of dimension $(M + 1)$.

Solving Eq. (14.130) for $\mathbf{w}'(n)$, we get

$$\mathbf{w}'(n) = r(n)\mathbf{R}'^{-1}(n) \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (14.131)$$

where $\mathbf{R}'^{-1}(n)$ is the inverse of matrix $\mathbf{R}'(n)$.

Equation (14.131) shows that $\mathbf{w}'(n)$ consists of the last column of $\mathbf{R}'^{-1}(n)$, scaled by the multiplier $r'(n)$. Indeed, this equation provides the basis of a processor for the parallel extraction of the least-squares weight vector $\hat{\mathbf{w}}(n)$, as depicted in Fig. 14.15. The systolic processor shown here is made up of the interconnection of two equidimensional components with distinct but coupled functions:

1. A standard triangular array processor, consisting of both boundary and internal cells. The input vector $\mathbf{u}(n)$ and the desired response $d(n)$ constitute the composite input for this basic part of the processor. The operation performed here is the recursive QR-decomposition of the augmented data matrix $\mathbf{A}'(n)$. In particular, the standard triangular array $\mathbf{R}'^H(n)$ looks after the updating of the corresponding lower triangular matrix $\mathbf{R}''^H(n)$, as shown by

$$[\mathbf{R}''^H(n), \mathbf{O}', \mathbf{0}] = [\lambda^{1/2}\mathbf{R}''^H(n - 1), \mathbf{V}'^T \begin{bmatrix} \mathbf{u}(n) \\ d(n) \end{bmatrix}] \mathbf{T}'^H(n) \quad (14.132)$$

where $\mathbf{T}'^H(n)$ denotes a sequence of $(M + 1)$ Givens rotations.

2. A triangular array postprocessor, consisting of internal cells only. The internal cells of this second array are fed with the appropriate rotation parameters computed in the boundary cells of the standard processor. The function of the postprocessor is to provide for the updating of $\mathbf{R}'^{-1}(n)$. In this context, we note from Eq. (14.111) that

$$\begin{bmatrix} \mathbf{R}'^{-H}(n) \\ \mathbf{X} \end{bmatrix} = \mathbf{T}'^H(n) \begin{bmatrix} \lambda^{-1/2}\mathbf{R}'^{-H}(n - 1) \\ \mathbf{0} \end{bmatrix} \quad (14.133)$$

where \mathbf{X} is a matrix of no specific interest.

The last boundary cell of the standard processor computes the multiplier $r'(n)$. The last (rightmost) column of internal cells in the postprocessor computes the last column of $\mathbf{R}'^{-1}(n - 1)$. Multiplication of the column vector so derived by the real

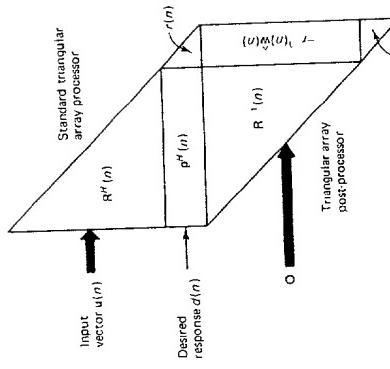


Figure 14.15 Systolic processor for parallel weight extraction

scalar $r'(n)$ yields $\hat{\mathbf{w}}(n)$ in accordance with Eq. (14.131). The first M elements of $\hat{\mathbf{w}}(n)$ define the desired least-squares weight vector $\hat{\mathbf{w}}(n)$, except for a minus sign. Note that the bottom cell in the last column of the postprocessor computes $1/r'(n)$; when this value is multiplied by $r'(n)$, the result is unity, which is in accord with the fact that the last element of $\hat{\mathbf{w}}(n)$ is unity. Checking that the last element of $\hat{\mathbf{w}}(n)$ is unity provides a certain measure of fault detection!

14.16 SUMMARY AND DISCUSSION

In this chapter, we have studied various systolic array structures for the recursive solution of both unconstrained and constrained versions of the linear least-squares problem. In this concluding section, we sum up the common features of these structures, their basic differences, and their areas of application.

To begin with, the systolic array structures described herein share the following common features:

1. They all exploit the QR-decomposition method that involves successive applications of a unitary transformation (i.e., the Givens rotation) directly to the data matrix $\mathbf{A}(n)$. The QR-decomposition method is known to be numerically stable, with the result that these structures are less susceptible to the effects of finite-precision arithmetic than would be the case if direct matrix inversion were to be applied to the correlation matrix $\Phi(n)$ in order to solve the system of normal equations $\Phi(n)\hat{\mathbf{w}}(n) = \mathbf{0}(n)$. Also, the QR-decomposition is updated recursively as new data arrive.
2. They are examples of an open-loop data-adaptive system, data adaptive in the sense that the values assigned to the parameters of the sequence of Givens ro-

8. The systolic arrays of Figs. 14.2 and 14.7 require the use of a common *clock signal* distributed to every cell (internal as well as boundary) in order to control data movements in the array. The need for synchronization will eventually become intolerable for very large scale or ultralarge-scale arrays. We may bypass the need for a common clock and therefore eliminate the synchronization problem by using the *wavefront array technique*⁷. A wavefront array processor implements the same basic algorithm as the systolic array. It takes advantage of the *control-flow locality* and *data-flow locality* that are inherently possessed by most algorithms of interest. These algorithmic features permit the use of a *data-driven, self-timed approach* to array processing. In effect, correct sequencing is substituted for the requirement of correct timing.
- Next we sum up the difference between these structures, and their special features:
3. In a systolic array, matrix computations are often directly *mapped* onto the processor array. This is, for example, readily seen by comparing the systolic array of Fig. 14.2 and governing equations (14.14) and (14.21).
 4. Systolic structures are relatively easy to design because of their highly pipelined operation, regular nature, and layout. These features, and the fact that only nearest-neighbor interconnections are involved between individual processing cells in the arrays, make them particularly well suited for implementation in VLSI form.
 5. The triangular systolic array that performs the sequence of Givens rotations consists of boundary cells, inner cells, and (in the case of Fig. 14.7) only a final processing cell. The number of boundary cells (representing the most complicated part of the algorithm in that square roots and divisions as well as multiplications and additions are performed there) equals $N - 1$, where N is the number of data points entering the array. The number of inner cells (where multiplications and additions only are performed), on the other hand, equals $(N^2 - N)/2$. For the structures of Figs. 14.2 and 14.7, we have $N = M + 1$, where M is the dimension of the input vector $\mathbf{u}(n)$ and the 1 accounts for the desired response $d(n)$. In Fig. 14.4 we have $N = M$. In addition to the triangular systolic array, the structure of Fig. 14.2 includes a linear systolic array for computing the N -by-1 least-squares weight vector $\hat{\mathbf{w}}(n)$; this section consists of one boundary cell (where addition, multiplication and division are performed) and $M - 1$ internal cells (where addition and multiplication are performed). The structure of Fig. 14.7 includes a final processing cell in the triangular systolic array, where a single multiplication is performed; this cell is used to compute the (a posteriori) estimation error $e'(n)$.
 6. The systolic arrays of Figs. 14.2 and 14.7 are *scalable* architecturally; that is, the size of the array may be enlarged indefinitely so long as the system synchronization is fully maintained.
 7. As a consequence of the highly pipelined nature of the systolic array structures in Figs. 14.2 and 14.7, and therefore the requirement to impose a time skew on their input data, there is an *overall delay* or *latency* in their response.

⁶With *very large scale integration* (VLSI), it is possible to build complex array processors that consist of a large number of processing cells, which cooperate with each other to perform extremely fast matrix computations. Kung (1985) presents an overview of VLSI array processors, with particular reference to systolic arrays. A major problem that arises from the use of this technology, however, is that a single flaw in the chip or wafer can render an entire VLSI array processor useless. It is therefore desirable to have a VLSI array processor that achieves high performance, tolerates physical failures, and yet produces correct results. Abraham et al. (1987) describe various techniques for fault tolerance that can be applied to systolic array architectures. A technique referred to as "algorithm-based fault tolerance" is shown to be a natural approach for these array processors. [see also Jou and Abramam (1986)] See also Aninson and Luk (1988) for fault-tolerant techniques for MVDR beamforming.

1. The structure of Fig. 14.2 solves the linear least-squares problem by producing the weight vector $\hat{\mathbf{w}}(n)$ at the output of the linear systolic array section. Such a structure may therefore find use in applications such as system identification where explicit knowledge of the weight vector is required.
2. The structure of Fig. 14.7 solves the linear least-squares problem by producing the a posteriori estimation error $e'(n)$ without computing the weight vector $\hat{\mathbf{w}}(n)$ explicitly. This structure achieves the highest possible throughput rate with the minimum amount of computation and circuitry. The structure of Fig. 14.7 may also have a numerical advantage over that of Fig. 14.2 since the former structure avoids the need to solve a triangular linear system, such a system of equations may sometimes be ill-conditioned.
3. The weight vector $\hat{\mathbf{w}}(n)$ may be extracted indirectly from the structure of Fig. 14.7 by means of a technique called "weight flushing." Moreover, it may be extended by the inclusion of a triangular array postprocessor for the (direct) parallel extraction of the weight vector $\hat{\mathbf{w}}(n)$.
4. The structure of Fig. 14.7 is very well suited for adaptive beamforming. Through the use of a constraint preprocessor, a linear constraint may be imposed on the operation of the processor so as to guard against the adaptive nulling of the target signal.
5. For MVDR beamforming, we may use a constrained extension of the structure of Fig. 14.2 or that of the structure of Fig. 14.7. The constraint is determined by the choice of a steering vector $s(\phi)$, where ϕ is some electrical angle. The structure of Fig. 14.2 remains invariant to this constraint. However, for every desired steering vector, we have to use a corresponding linear systolic array to perform the pertinent back substitution. This difficulty may be resolved by using the structure of Fig. 14.7 and building on it in the manner described in Section 14.13. In so doing, only one systolic array is required to solve the MVDR beamforming problem for the many look directions of interest.

⁷The wavefront array technique was originated by S. Y. Kung. For a tutorial treatment of it, see the review papers Kung, 1985; Kung et al., 1987 [see also the book Kung, 1988]. For the use of a wavefront array processor for adaptive beamforming, see McCanny and McWhirter (1987).

Comparison of RLS, LMS, and Sign Algorithms Tracking Randomly Time-Varying Channels

Eweda Eweda, *Senior Member, IEEE*

J1017 U.S. PRO
09/911764
07/25/01

Abstract—The performance of adaptive FIR filters governed by the recursive least-squares (RLS) algorithm, the least mean square (LMS) algorithm, and the sign algorithm (SA), are compared when the optimal filtering vector is randomly time-varying. The comparison is done in terms of the steady-state excess mean-square estimation error ξ and the steady-state mean-square weight deviation, η . It is shown that ξ does not depend on the spread of eigenvalues of the input covariance matrix, R , in the cases of the LMS algorithm and the SA, while it does in the case of the RLS algorithm. In the three algorithms, η is found to be increasing with the eigenvalue spread. The value of the adaptation parameter that minimizes ξ is different from the one that minimizes η . It is shown that the minimum values of ξ and η attained by the RLS algorithm are equal to the ones attained by the LMS algorithm in any one of the three following cases: (1) if R has equal eigenvalues, (2) if the fluctuations of the individual elements of the optimal vector are mutually uncorrelated and have the same mean-square value, or (3) if R is diagonal and the fluctuations of the individual elements of the optimal vector have the same mean-square value. Conditions that make the values of ξ and η of the LMS algorithm smaller (or greater) than the ones of the RLS algorithm are derived. For Gaussian input data, the minimum values of ξ and η attained by the SA are found to exceed the ones attained by the LMS algorithm by 1 dB independently of R and the mutual correlation between the elements of the optimal vector.

I. INTRODUCTION

THE performance of an adaptive filter depends mainly on the algorithm used for updating the filter weights. Three well-known algorithms are: the recursive least-squares (RLS) algorithm [1]–[5], the least mean-square (LMS) algorithm [6], [10]–[12], and the sign algorithm (SA) [7]–[9]. Convergence analyses of the three algorithms in the case of stationary adaptive filtering lead to the following conclusions. With appropriate choice of the adaptation parameter, the RLS and LMS algorithms are exponentially convergent [5], [10] whereas the SA is linearly convergent [8]. This makes the convergence of the SA the slowest when the initial weight setting is far from the optimum. The convergence rate of the LMS algorithm decreases as the spread of eigenvalues of the input covariance matrix increases [11] whereas the convergence rate of the RLS algorithm is independent of the eigenvalue spread [5].

The purpose of this paper is to give a quantitative comparison of the three algorithms when tracking a randomly

Manuscript received March 19, 1993; revised March 1, 1994. The associate editor coordinating the review of this paper and approving it for publication was Dr. Fuyun Ling.

The author is with the Military Technical College, Kobry El-Kobba, Cairo, Egypt.

IEEE Log Number 9404766.

time-varying channel. While a reasonable progress has been achieved in the tracking analysis of the LMS algorithm and the SA (e.g., [6], [7], [9], [12]), the available tracking analysis of the RLS algorithm [1]–[3], [14] is far from being complete. In [1], the RLS algorithm is studied when the signal characteristic has a step change, and it is shown that the algorithm converges exponentially with a time constant equal to one half of the reciprocal of the adaptation parameter. In [2], the RLS algorithm is studied in the case when the fluctuations of the individual channel parameters are Gaussian, mutually uncorrelated, and of the same mean-square value. Under these assumptions, it is shown [2, p. 1102] that the RLS algorithm has the same tracking capability as the LMS algorithm when the input covariance matrix has equal eigenvalues. In [3], it is shown that the performance of the RLS algorithm is worse than that of the LMS algorithm in the case of tracking a deterministic chirped sinusoid buried in additive white noise.

The present paper provides a tracking analysis of the RLS algorithm that does not include assumptions on the type of distribution and mutual correlation among the channel parameters. The paper compares the RLS, LMS, and SA algorithms from the points of view of the steady-state excess mean-square estimation error ξ and the steady-state mean-square weight deviation, η . As will be shown in the paper, the value of the adaptation parameter μ , that minimizes ξ , is generally different from the one that minimizes η . Thus, the optimum value of μ depends on whether ξ or η is adopted as the primary performance index. The paper derives expressions of η for the three algorithms and generalizes previously derived expressions of ξ . For each algorithm, the value of μ that minimizes η and the one that minimizes ξ are derived. The paper compares the three algorithms on the base of the minimum values of η and ξ attained by each algorithm. Conditions that make the tracking properties of the LMS algorithm better (or worse) than those of the RLS algorithm are derived. The paper is organized as follows. Section II is concerned with the problem formulation and the assumptions used throughout the paper. Sections III–V are concerned with the derivation of ξ and η of the RLS algorithm, the LMS algorithm, and the sign algorithm, respectively. Section VI compares the minimum values of ξ and η of the three algorithms and provides a decision tree for the choice between the RLS and LMS algorithms. Section VII provides computer simulations that support the derived theoretical results. The conclusions of the paper are given in Section VIII.

II. PROBLEM FORMULATION

Let \underline{h}_k , \underline{x}_k , and a_k denote the weight vector of the adaptive filter at discrete time k , the observation vector, and the desired filter output, respectively. The estimation error e_k is given by

$$e_k \triangleq a_k - \underline{h}_k^T \underline{x}_k \quad (2.1)$$

where \underline{h}_k^T is the transpose of \underline{h}_k . The RLS algorithm is given by [1]–[5]

$$\underline{h}_{k+1} = \underline{h}_k + \mu P_{k+1} \underline{x}_k e_k, \quad 0 < \mu < 1 \quad (2.2)$$

$$P_{k+1} = \frac{1}{1-\mu} \left[P_k - \frac{\mu P_k \underline{x}_k \underline{x}_k^T P_k}{1-\mu + \mu \underline{x}_k^T P_k \underline{x}_k} \right]. \quad (2.3)$$

In (2.3), P_1 is an arbitrary symmetric positive definite matrix. The quantity $(1-\mu)$ is usually referred to as the forgetting factor of the RLS algorithm. The LMS algorithm is given by [6]

$$\underline{h}_{k+1} = \underline{h}_k + \mu \underline{x}_k e_k, \quad 0 < \mu < \mu_0 \quad (2.4)$$

where μ_0 is a positive number that depends on the statistics of \underline{x}_k . Finally, the SA is given by [7]–[9]

$$\underline{h}_{k+1} = \underline{h}_k + \mu \underline{x}_k \operatorname{sgn}(e_k), \quad \mu > 0. \quad (2.5)$$

Let the relation between the desired filter output a_k and the observation vector, \underline{x}_k , be modeled by

$$a_k = \underline{c}_k^T \underline{x}_k + b_k \quad * \quad (2.6)$$

where \underline{c}_k and b_k are the model parameter vector and the model noise, respectively. Examples of adaptive filtering applications that follow the model (2.6) are adaptive system identification and adaptive echo canceling [15]. The vector \underline{c}_k is assumed randomly time-varying. Let the increment of \underline{c}_k be denoted by

$$\underline{d}_k \triangleq \underline{c}_{k+1} - \underline{c}_k. \quad (2.7)$$

The following assumptions are used throughout the paper:

- A.1) The sequences $\{\underline{x}_k\}$, $\{b_k\}$, and $\{\underline{d}_k\}$ are mutually independent.
- A.2) $\{\underline{d}_k\}$ is a stationary sequence of independent zero-mean vectors, and $\underline{Q} \triangleq E(\underline{d}_k \underline{d}_k^T)$ is the covariance matrix of the increments.
- A.3) The sequence $\{\underline{x}_k\}$ is stationary, zeromean, and the covariance matrix $\underline{R} \triangleq E(\underline{x}_k \underline{x}_k^T)$ is positive definite.
- A.4) $\{b_k\}$ is a stationary sequence of independent zeromean random variables with variance σ_b^2 .

These assumptions will be used to analyze the tracking of the RLS and LMS algorithms. The tracking analysis of the SA will use the assumptions (A.1)–(A.4) along with the assumption that the sequences $\{b_k\}$ and $\{\underline{x}_k\}$ are Gaussian. These assumptions are the same as the ones considered in [7], [9], and [13]. Another model of variation of \underline{c}_k that has been considered in the context of nonstationary adaptive filtering is a Markovian model [6] that can be expressed as

model is close to the model expressed by the assumption (A.2). Indeed, it is easy to show that this model satisfies

$$|E(\underline{d}_{k+1}^T \underline{d}_k)| / |E(\underline{d}_k^T \underline{d}_k)| = (1-\alpha)/2 \approx 0$$

which means that \underline{d}_k and \underline{d}_{k+1} are nearly uncorrelated. It should be noticed that the assumptions (A.1)–(A.4) do not imply that the individual elements of \underline{x}_k have the same mean-square value. Therefore, (A.1)–(A.4) allow the case when the individual elements of \underline{x}_k are derived from different signals as well as the case when \underline{x}_k is made of samples of the same signal.

The steady-state excess mean-square estimation error ξ is defined by

$$\xi \triangleq \lim_{k \rightarrow \infty} E(e_k^2) - \sigma_b^2. \quad (2.8)$$

Let the instantaneous weight deviation from the optimal be denoted by

$$\underline{v}_k \triangleq \underline{h}_k - \underline{c}_k. \quad (2.9)$$

The steady-state mean-square weight deviation η is defined by

$$\eta \triangleq \lim_{k \rightarrow \infty} E(\|\underline{v}_k\|^2) \quad (2.10)$$

where $\|\underline{v}_k\|^2 \triangleq \underline{v}_k^T \underline{v}_k$ is the squared norm. In the following sections, we derive expressions of ξ and η for each algorithm.

III. TRACKING PROPERTIES OF THE RLS ALGORITHM

Denote

$$\underline{R}_k \triangleq \underline{P}_k^{-1}. \quad (3.1)$$

Equations (2.3), (3.1), and the matrix inversion lemma imply

$$\boxed{\underline{R}_{k+1} = (1-\mu)\underline{R}_k + \mu \underline{x}_k \underline{x}_k^T.} \quad (3.2)$$

From (2.1), (2.2), (2.6), (2.7), (2.9), and (3.1) one obtains

$$\underline{v}_{k+1} = \underline{v}_k + \mu \underline{R}_{k+1}^{-1} \underline{x}_k (b_k - \underline{v}_k^T \underline{x}_k) - \underline{d}_k. \quad (3.3)$$

Premultiplying both sides of (3.3) by \underline{R}_{k+1} and then using (3.2), one obtains

$$\underline{R}_{k+1} \underline{v}_{k+1} = (1-\mu) \underline{R}_k \underline{v}_k + \mu b_k \underline{x}_k - \underline{R}_{k+1} \underline{d}_k. \quad (3.4)$$

Due to (3.2), the sequence $\{\underline{R}_k\}$ depends only on the sequence $\{\underline{x}_k\}$. Then (A.1) implies that $\{\underline{R}_k\}$ is independent of the sequences $\{b_k\}$ and $\{\underline{d}_k\}$. Due to (3.3), \underline{v}_k is a function of $\{b_j, \underline{d}_j, \underline{x}_j\}; j < k\}$. Therefore, (A.1), (A.2), and (A.4) imply that b_k and \underline{d}_k are independent of \underline{v}_k . Squaring (3.4), using the independence of b_k and \underline{d}_k on $\underline{v}_k, \underline{x}_k, \underline{R}_k$, and \underline{R}_{k+1} , using the fact that b_k and \underline{d}_k are zeromean and that $E(b_k^2) = \sigma_b^2$ and $E(\underline{d}_k \underline{d}_k^T) = \underline{Q}$ imply that

$$\begin{aligned} & E(\underline{R}_{k+1} \underline{v}_{k+1} \underline{v}_{k+1}^T \underline{R}_{k+1}) \\ &= (1-\mu)^2 E(\underline{R}_k \underline{v}_k \underline{v}_k^T \underline{R}_k) + \mu^2 \sigma_b^2 \underline{R} + E(\underline{R}_{k+1} \underline{Q} \underline{R}_{k+1}). \end{aligned} \quad (3.5)$$

$$\frac{1}{(\mu-\lambda)^2} \approx \frac{1}{N(\mu-\lambda)}$$

Therefore, for small values of μ , we can use the approximation that $\underline{R}_k = \underline{R}$ in the steady state. In such a case, for large values of k , (3.5) can be approximated by

$$\begin{aligned} & \underline{R}E(\underline{v}_{k+1}\underline{v}_{k+1}^T)\underline{R} \\ &= (1-\mu)^2 \underline{R}E(\underline{v}_k\underline{v}_k^T)\underline{R} + \mu^2 \sigma_b^2 \underline{R} + \underline{R}Q\underline{R}. \end{aligned} \quad (3.7)$$

Premultiplying and postmultiplying the two sides of (3.7) by \underline{R}^{-1} yields

$$E(\underline{v}_{k+1}\underline{v}_{k+1}^T) = (1-\mu)^2 E(\underline{v}_k\underline{v}_k^T) + \mu^2 \sigma_b^2 \underline{R}^{-1} + \underline{Q}. \quad (3.8)$$

Since $0 < \mu < 1$, then (3.8) implies that

$$\lim_{k \rightarrow \infty} E(\underline{v}_k\underline{v}_k^T) = \frac{1}{2-\mu} [\mu \sigma_b^2 \underline{R}^{-1} + \mu^{-1} \underline{Q}]. \quad (3.9)$$

The steady-state mean-square weight deviation is then given by

$$\begin{aligned} \eta^{\text{RLS}} &\triangleq \lim_{k \rightarrow \infty} E(\|\underline{v}_k\|^2) \\ &= \frac{1}{2-\mu} [\mu \sigma_b^2 \text{tr}(\underline{R}^{-1}) + \mu^{-1} \text{tr}(\underline{Q})] \end{aligned} \quad (3.10)$$

where $\text{tr}(\cdot)$ denotes the trace of (\cdot) . The first term on the RHS of (3.10) is due to the plant noise while the second one is due to the plant nonstationarity. The first term is increasing in μ whereas the second one is decreasing in μ ; $0 < \mu < 1$. Assuming that $\mu \ll 1$, as is usually the case in applications, then the value of μ , denoted by μ_η^{RLS} , that minimizes η^{RLS} , and the minimum value of η^{RLS} , denoted by η_{\min}^{RLS} , are, respectively, given by

$$\mu_\eta^{\text{RLS}} = \left[\frac{\text{tr}(\underline{Q})}{\sigma_b^2 \text{tr}(\underline{R}^{-1})} \right]^{1/2} \quad (3.11)$$

$$\eta_{\min}^{\text{RLS}} = \sigma_b \sqrt{\text{tr}(\underline{Q}) \text{tr}(\underline{R}^{-1})}. \quad (3.12)$$

Now, to find the steady-state mean-square estimation error ξ^{RLS} , we shall use the following assumption:

A.5) \underline{v}_k and \underline{x}_k are statistically independent.

This assumption has been used by many authors (e.g., [7], [8], [13]) and is approximately valid at small values of μ . Due to (2.1), (2.6), and (2.9), $e_k = b_k - \underline{v}_k^T \underline{x}_k$. Then (A.5) and the fact that b_k is zeromean and independent of $\underline{v}_k^T \underline{x}_k$ yield

$$E(e_k^2) = \sigma_b^2 + E(\underline{x}_k^T E(\underline{v}_k \underline{v}_k^T) \underline{x}_k). \quad (3.13)$$

Equations (2.8), (3.9), and (3.13) imply that the steady-state excess mean-square error of the RLS algorithm is given by

$$\begin{aligned} \xi^{\text{RLS}} &= \frac{1}{2-\mu} [\mu \sigma_b^2 E(\underline{x}_k^T \underline{R}^{-1} \underline{x}_k) + \mu^{-1} E(\underline{x}_k^T \underline{Q} \underline{x}_k)] \\ &= \frac{1}{2-\mu} [N \mu \sigma_b^2 + \mu^{-1} \text{tr}(\underline{Q} \underline{R})] \end{aligned} \quad (3.14)$$

where N is the dimension of \underline{x}_k . The first term on the RHS of (3.14) is due to the plant noise while the second one is due to the plant nonstationarity. In the particular case when $\underline{Q} = c \underline{I}$, with \underline{I} being the identity matrix and c being an arbitrary positive number, the result (3.14) reduces to the one obtained in [2, p. 1102]. From (3.14), assuming that $\mu \ll 1$,

the value of μ that minimizes ξ^{RLS} , and the minimum value of ξ^{RLS} , are, respectively, given by

$$\mu_\xi^{\text{RLS}} = \left[\frac{\text{tr}(\underline{Q} \underline{R})}{N \sigma_b^2} \right]^{1/2} \quad (3.15)$$

$$\xi_{\min}^{\text{RLS}} = \sigma_b \sqrt{N \text{tr}(\underline{Q} \underline{R})}. \quad (3.16)$$

From (3.11) and (3.15), it is obvious that μ_ξ^{RLS} is different from μ_η^{RLS} . In the particular case when $\underline{R} = c \underline{I}$, $\mu_\xi^{\text{RLS}} = \mu_\eta^{\text{RLS}}$. When $\underline{Q} = c \underline{I}$, $\mu_\xi^{\text{RLS}} = \sqrt{c \text{tr}(\underline{R}) / (N \sigma_b^2)}$ and $\mu_\eta^{\text{RLS}} = \sqrt{c N / (\sigma_b^2 \text{tr}(\underline{R}^{-1}))}$. In the latter case, μ_ξ^{RLS} does not depend on the eigenvalue spread of \underline{R} while μ_η^{RLS} tends to zero as the minimum eigenvalue of \underline{R} tends to zero.

IV. TRACKING PROPERTIES OF THE LMS ALGORITHM

Under the assumptions (A.1)–(A.5), and with μ satisfying $\mu \text{tr}(\underline{R}) \ll 1$, it has been shown in [7, p. 2053] that the steady-state excess mean-square error of the LMS algorithm is given by

$$\xi^{\text{LMS}} = \frac{1}{2} [\mu \sigma_b^2 \text{tr}(\underline{R}) + \mu^{-1} \text{tr}(\underline{Q})]. \quad (4.1)$$

In the particular case when $\underline{Q} = c \underline{I}$, (4.1) reduces to the result obtained in [6]. Due to (4.1), the value of μ that minimizes ξ^{LMS} , and the minimum value of ξ^{LMS} , are, respectively, given by

$$\mu_\xi^{\text{LMS}} = \left[\frac{\text{tr}(\underline{Q})}{\sigma_b^2 \text{tr}(\underline{R})} \right]^{1/2} \quad (4.2)$$

$$\xi_{\min}^{\text{LMS}} = \sigma_b \sqrt{\text{tr}(\underline{R}) \text{tr}(\underline{Q})}. \quad (4.3)$$

In order that μ_ξ^{LMS} satisfy the condition $\mu_\xi^{\text{LMS}} \text{tr}(\underline{R}) \ll 1$, one should have

$$\sqrt{\text{tr}(\underline{R}) \text{tr}(\underline{Q})} \ll \sigma_b. \quad (4.4)$$

This condition is usually called the condition of slow variations [12].

In the following, we derive the steady-state mean-square weight deviation, η^{LMS} . From (2.1), (2.4), (2.6), (2.7), and (2.9) one has

$$\underline{v}_{k+1} = (\underline{I} - \mu \underline{x}_k \underline{x}_k^T) \underline{v}_k + \mu b_k \underline{x}_k - \underline{d}_k. \quad (4.5)$$

Squaring (4.5) and using the assumptions (A.1)–(A.5) yield

$$E(\underline{v}_{k+1}\underline{v}_{k+1}^T) = \underline{E}_k + \mu^2 \sigma_b^2 \underline{R} + \underline{Q} \quad (4.6)$$

where

$$\begin{aligned} \underline{E}_k &= E[(\underline{I} - \mu \underline{x}_k \underline{x}_k^T) E(\underline{v}_k \underline{v}_k^T) (\underline{I} - \mu \underline{x}_k \underline{x}_k^T)] \\ &= E(\underline{v}_k \underline{v}_k^T) - \mu [E(\underline{R} E(\underline{v}_k \underline{v}_k^T) + E(\underline{v}_k \underline{v}_k^T) \underline{R}) \\ &\quad + \mu^2 E(\underline{x}_k \underline{x}_k^T E(\underline{v}_k \underline{v}_k^T) \underline{x}_k \underline{x}_k^T)]. \end{aligned} \quad (4.7)$$

For small μ , the last term on the right-hand side of (4.7) can be neglected with respect to the second term and then (4.6) and (4.7) yield

$$\begin{aligned} & E(\underline{v}_{k+1}\underline{v}_{k+1}^T) \\ &= E(\underline{v}_k \underline{v}_k^T) - \mu [E(\underline{R} E(\underline{v}_k \underline{v}_k^T) + E(\underline{v}_k \underline{v}_k^T) \underline{R}) + \mu^2 \sigma_b^2 \underline{R} + \underline{Q}]. \end{aligned} \quad (4.8)$$

In the steady state, $E(\underline{v}_{k+1}\underline{v}_{k+1}^T) = E(\underline{v}_k\underline{v}_k^T)$ and then (4.8) gives

$$\lim_{k \rightarrow \infty} [RE(\underline{v}_k\underline{v}_k^T) + E(\underline{v}_k\underline{v}_k^T)R] = \mu\sigma_b^2\underline{R} + \mu^{-1}\underline{Q}. \quad (4.9)$$

It should be noticed that (4.1) can be obtained by taking the trace of both sides of (4.9). Now, since the covariance matrix \underline{R} is symmetric and positive definite, it can be decomposed as

$$\underline{R} = \underline{U}\underline{\Lambda}\underline{U}^T \quad (4.10)$$

where \underline{U} is an orthonormal matrix and

$$\underline{\Lambda} \triangleq \text{diag}(\lambda_1, \dots, \lambda_N) \quad (4.11)$$

with $\lambda_1, \dots, \lambda_N$ being the eigenvalues of \underline{R} . Premultiplying and postmultiplying both sides of (4.9) by \underline{U}^T and \underline{U} , respectively, yields

$$\underline{\Lambda}\underline{S}' + \underline{S}'\underline{\Lambda} = \mu\sigma_b^2\underline{\Lambda} + \mu^{-1}\underline{Q}' \quad (4.12)$$

where

$$\underline{Q}' \triangleq \underline{U}^T \underline{Q} \underline{U} \quad (4.13)$$

$$\underline{S}' \triangleq \underline{U}^T \underline{S} \underline{U} \quad (4.14)$$

$$\underline{S} \triangleq \lim_{k \rightarrow \infty} E(\underline{v}_k\underline{v}_k^T). \quad (4.15)$$

Postmultiplying both sides of (4.12) by $\underline{\Lambda}^{-1}$ and then taking the trace of both sides yield

$$\text{tr}(\underline{S}') = \frac{1}{2}[\mu N\sigma_b^2 + \mu^{-1} \text{tr}(\underline{Q}'\underline{\Lambda}^{-1})]. \quad (4.16)$$

Due to (4.10) and (4.13), $\text{tr}(\underline{Q}'\underline{\Lambda}^{-1}) = \text{tr}(\underline{Q}\underline{R}^{-1})$, and then (4.14)–(4.16) imply that the steady-state mean-square weight deviation of the LMS algorithm is given by

$$\left\{ \begin{array}{l} \eta^{\text{LMS}} \triangleq \lim_{k \rightarrow \infty} E(\|\underline{v}_k\|^2) \\ = \frac{1}{2}[\mu N\sigma_b^2 + \mu^{-1} \text{tr}(\underline{Q}\underline{R}^{-1})]. \end{array} \right\} \quad (4.17)$$

The value of μ that minimizes η^{LMS} and the minimum value of η^{LMS} are, respectively, given by

$$\mu_\eta^{\text{LMS}} = \sqrt{\text{tr}(\underline{Q}\underline{R}^{-1})/(N\sigma_b^2)} \quad (4.18)$$

$$\eta_{\min}^{\text{LMS}} = \sigma_b \sqrt{N \text{tr}(\underline{Q}\underline{R}^{-1})}. \quad (4.19)$$

From (4.2) and (4.18), it is seen that μ_ξ^{LMS} does not depend on the eigenvalue spread of \underline{R} while μ_η^{LMS} does. In the particular case when $\underline{R} = cI$, $\mu_\xi^{\text{LMS}} = \mu_\eta^{\text{LMS}}$.

V. TRACKING PROPERTIES OF THE SA

The analysis given in Section III and IV does not include any assumption on the type of distribution of \underline{x}_k and b_k . In

and the corresponding optimum value of μ are, respectively, given by

$$\begin{aligned} \xi_{\min}^{\text{SA}} &= \frac{\pi}{4} \text{tr}(\underline{R}) \text{tr}(\underline{Q}) \\ &+ \left[\frac{\pi}{2} \sigma_b^2 \text{tr}(\underline{R}) \text{tr}(\underline{Q}) + \frac{\pi^2}{16} \text{tr}^2(\underline{R}) \text{tr}^2(\underline{Q}) \right]^{1/2} \end{aligned} \quad (5.1)$$

$$\mu_\xi^{\text{SA}} = \sqrt{\text{tr}(\underline{Q})/\text{tr}(\underline{R})}. \quad (5.2)$$

Equation (5.2) is the same as the one derived in [9]. Under the condition (4.4) of slow variations, (5.1) reduces to

$$\xi_{\min}^{\text{SA}} = \sigma_b \sqrt{\pi \text{tr}(\underline{R}) \text{tr}(\underline{Q})/2}. \quad (5.3)$$

It has also been shown in [7, (51)] that

$$\sqrt{2/\pi} \mu [\underline{S}' + \underline{\Lambda}\underline{S}'\underline{\Lambda}^{-1}] / \sigma_b = \mu^2 I + \underline{Q}'\underline{\Lambda}^{-1} \quad (5.4)$$

where $\underline{\Lambda}$, \underline{Q}' , and \underline{S}' are defined by (4.11), (4.13), and (4.14), respectively. Taking the trace of both sides of (5.4) and using (4.14), then

$$\sqrt{8/\pi} \mu \text{tr}(\underline{S}) / \sigma_b = N\mu^2 + \text{tr}(\underline{Q}'\underline{\Lambda}^{-1}). \quad (5.5)$$

Due to (4.10) and (4.13), $\text{tr}(\underline{Q}'\underline{\Lambda}^{-1}) = \text{tr}(\underline{Q}\underline{R}^{-1})$. Then (5.5) and (4.15) imply that the steady-state mean-square weight deviation of the SA is given by

$$\begin{aligned} \eta^{\text{SA}} &\triangleq \lim_{k \rightarrow \infty} E(\|\underline{v}_k\|^2) \\ &= \sqrt{\pi/8} \sigma_b [\mu N + \mu^{-1} \text{tr}(\underline{Q}\underline{R}^{-1})]. \end{aligned} \quad (5.6)$$

Then the value of μ that minimizes η^{SA} and the minimum value of η^{SA} are, respectively, given by

$$\mu_\eta^{\text{SA}} = \sqrt{\text{tr}(\underline{Q}\underline{R}^{-1})/N} \quad (5.7)$$

$$\eta_{\min}^{\text{SA}} = \sigma_b \sqrt{\pi N \text{tr}(\underline{Q}\underline{R}^{-1})/2}. \quad (5.8)$$

From (5.2) and (5.7), it is seen that μ_ξ^{SA} does not depend on the eigenvalue spread of \underline{R} while μ_η^{SA} does. In the particular case when $\underline{R} = cI$, $\mu_\xi^{\text{SA}} = \mu_\eta^{\text{SA}}$. It is worth mentioning that, due to (4.2), (4.18), (5.2), and (5.7), the optimum values of μ of the LMS algorithm and the SA are related to each other according to

$$\mu_\xi^{\text{SA}} = \sigma_b \mu_\xi^{\text{LMS}}, \quad \mu_\eta^{\text{SA}} = \sigma_b \mu_\eta^{\text{LMS}}. \quad (5.9)$$

VI. COMPARISON OF THE TRACKING PROPERTIES OF THE ALGORITHMS

Table I summarizes the above derived minimum mean-square errors, minimum mean-square weight deviations, and optimum adaptation parameters of the three algorithms.¹ From this table, one observes that ξ_{\min}^{LMS} , μ_ξ^{LMS} , ξ_{\min}^{SA} , and μ_ξ^{SA} do not depend on the spread of eigenvalues of \underline{R} , whereas ξ_{\min}^{RLS} and μ_ξ^{RLS} do. On the other hand, μ_η^{LMS} and μ_η^{SA} of the

TABLE I
MINIMUM MEAN-SQUARE ERROR, MINIMUM MEAN-SQUARE WEIGHT DEVIATION, AND OPTIMUM ADAPTATION PARAMETERS OF THE RLS ALGORITHM, LMS ALGORITHM, AND SA

	RLS Algorithm	LMS Algorithm	Sign Algorithm
ξ_{\min}	$\sigma_b \sqrt{N \text{tr}(\underline{Q}\underline{R})}$	$\sigma_b \sqrt{\text{tr}(\underline{R}) \text{tr}(\underline{Q})}$	$\sigma_b \sqrt{\pi \text{tr}(\underline{R}) \text{tr}(\underline{Q})/2}$
η_{\min}	$\sigma_b \sqrt{\text{tr}(\underline{Q}) \text{tr}(\underline{R}^{-1})}$	$\sigma_b \sqrt{N \text{tr}(\underline{Q}\underline{R}^{-1})}$	$\sigma_b \sqrt{n N \text{tr}(\underline{Q}\underline{R}^{-1})/2}$
μ_ξ	$\left[\frac{\text{tr}(\underline{Q}\underline{R})}{N \sigma_b^2} \right]^{1/2}$	$\left[\frac{\text{tr}(\underline{Q})}{\sigma_b^2 \text{tr}(\underline{R})} \right]^{1/2}$	$\sqrt{\text{tr}(\underline{Q})/\text{tr}(\underline{R})}$
μ_η	$\left[\frac{\text{tr}(\underline{Q})}{\sigma_b^2 \text{tr}(\underline{R}^{-1})} \right]^{1/2}$	$\sqrt{\text{tr}(\underline{Q}\underline{R}^{-1})/(N \sigma_b^2)}$	$\sqrt{\text{tr}(\underline{Q}\underline{R}^{-1})/N}$

weight deviation increases as the minimum eigenvalue of \underline{R} decreases. From Table I one has

$$\frac{\xi_{\min}^{\text{RLS}}}{\xi_{\min}^{\text{LMS}}} = \left[\frac{N \text{tr}(\underline{Q}\underline{R})}{\text{tr}(\underline{R}) \text{tr}(\underline{Q})} \right]^{1/2} \quad (6.1)$$

and

$$\frac{\eta_{\min}^{\text{RLS}}}{\eta_{\min}^{\text{LMS}}} = \left[\frac{\text{tr}(\underline{Q}) \text{tr}(\underline{R}^{-1})}{N \text{tr}(\underline{Q}\underline{R}^{-1})} \right]^{1/2}. \quad (6.2)$$

From (6.1) and (6.2), $\xi_{\min}^{\text{RLS}} = \xi_{\min}^{\text{LMS}}$ and $\eta_{\min}^{\text{RLS}} = \eta_{\min}^{\text{LMS}}$ in any one of the three following cases: (1) if $\underline{R} = c\underline{I}$, with c being an arbitrary positive number, (2) if $\underline{Q} = c\underline{I}$, or (3) if \underline{R} is diagonal and the diagonal elements of \underline{Q} are equal. In these three cases, the LMS algorithm is more recommended than the RLS algorithm due to the technical simplicity of the former.

From (6.1), ξ_{\min}^{RLS} may be greater or smaller than ξ_{\min}^{LMS} , depending on whether $N \text{tr}(\underline{Q}\underline{R})$ is greater or smaller than $\text{tr}(\underline{Q}) \text{tr}(\underline{R})$, respectively. Similarly, from (6.2), η_{\min}^{RLS} may be greater or smaller than η_{\min}^{LMS} , depending on whether $\text{tr}(\underline{Q}) \text{tr}(\underline{R}^{-1})$ is greater or smaller than $N \text{tr}(\underline{Q}\underline{R}^{-1})$, respectively. In the following, we give a geometrical meaning of these conditions. Let $\lambda_1, \lambda_2, \dots, \lambda_N; \lambda_1 > \lambda_2 > \dots > \lambda_N > 0$ be the eigenvalues of the matrix \underline{R} , let $\underline{u}_1, \underline{u}_2, \dots, \underline{u}_N$ be the associated eigenvectors, respectively, and let

$$\gamma_i \triangleq E((\underline{d}_k^T \underline{u}_i)^2), \quad i = 1, 2, \dots, N \quad (6.3)$$

denote the mean square of the projection of \underline{d}_k along the direction of \underline{u}_i . In the appendix, we show that

$$\begin{aligned} & N \text{tr}(\underline{Q}\underline{R}) - \text{tr}(\underline{Q}) \text{tr}(\underline{R}) \\ &= \sum_{i=1}^{N-1} \sum_{j=i+1}^N (\lambda_i - \lambda_j)(\gamma_i - \gamma_j) \end{aligned} \quad (6.4)$$

$$\begin{aligned} & N \text{tr}(\underline{Q}\underline{R}^{-1}) - \text{tr}(\underline{Q}) \text{tr}(\underline{R}^{-1}) \\ &= \sum_{i=1}^{N-1} \sum_{j=i+1}^N (\lambda_i^{-1} - \lambda_j^{-1})(\gamma_i - \gamma_j). \end{aligned} \quad (6.5)$$

Since $\lambda_i > \lambda_j$ for $i < j$, then due to (6.1), (6.2), (6.4), and (6.5), a sufficient condition of ξ_{\min}^{RLS} and η_{\min}^{RLS} to be

greater (resp. smaller) than ξ_{\min}^{LMS} and η_{\min}^{LMS} , respectively, is that $\gamma_i > \gamma_j$ (resp. $\gamma_i < \gamma_j$) for $i < j$. Thus, a sufficient condition for the LMS algorithm tracking to be better (resp. worse) than the RLS algorithm tracking is that the mean-square projection of the optimal vector increment along the direction of an eigenvector of \underline{R} be increasing (resp. decreasing) with the associated eigenvalue. The result matches the well-known fact [11] that the adaptation speed of the LMS algorithm along the direction of an eigenvector of \underline{R} is increasing with the associated eigenvalue, while the adaptation speed of the RLS algorithm is uniform with respect to the direction [1], [2], [5].

The above conditions can be used to choose between the RLS and LMS algorithms when the fluctuation covariance matrix \underline{Q} is known. Now, we derive a condition that can be used when \underline{Q} is unknown. Since $\text{tr}(\underline{Q}\underline{R}) = E(\underline{d}_k^T \underline{R}\underline{d}_k)$ and $\text{tr}(\underline{Q}) = E(\underline{d}_k^T \underline{d}_k)$, then $\lambda_{\min} \text{tr}(\underline{Q}) \leq \text{tr}(\underline{Q}\underline{R}) \leq \lambda_{\max} \text{tr}(\underline{Q})$ where λ_{\min} and λ_{\max} , respectively, denote the minimum and maximum eigenvalues of \underline{R} . Then, (6.1) implies that

$$\sqrt{N \lambda_{\min} / \text{tr}(\underline{R})} \leq \frac{\xi_{\min}^{\text{RLS}}}{\xi_{\min}^{\text{LMS}}} \leq \sqrt{N \lambda_{\max} / \text{tr}(\underline{R})}. \quad (6.6)$$

Similarly, (6.2) implies that

$$\sqrt{\lambda_{\min} \text{tr}(\underline{R}^{-1}) / N} \leq \frac{\eta_{\min}^{\text{RLS}}}{\eta_{\min}^{\text{LMS}}} \leq \sqrt{\lambda_{\max} \text{tr}(\underline{R}^{-1}) / N}. \quad (6.7)$$

The lower (resp. upper) bounds in (6.6) and (6.7) are attained when the parameter fluctuation vector \underline{d}_k has the same direction as the eigenvector associated with the minimum (resp. maximum) eigenvalue. Equations (6.6) and (6.7) can be used to choose between the RLS and the LMS algorithms when the matrix \underline{Q} is unknown as follows. When ξ is the primary performance index, it is reasonable to choose the LMS algorithm when the geometric mean of the lower and upper bounds in (6.6) is greater than or equal to 1, i.e., to use the LMS algorithm if $N \sqrt{\lambda_{\min} \lambda_{\max}} \geq \text{tr}(\underline{R})$ and to use the RLS algorithm otherwise. Similarly, when η is the primary performance index, it is recommended to use the LMS algorithm if $\text{tr}(\underline{R}^{-1}) \sqrt{\lambda_{\min} \lambda_{\max}} \geq N$ and to use the RLS algorithm otherwise. The above arguments about the choice between the RLS and LMS algorithms when \underline{Q} is known and when \underline{Q} is unknown are summarized in the decision tree given in Fig. 1.

Finally, from Table I, for Gaussian b_k and \underline{x}_k and for all values of \underline{R} and \underline{Q} , one has

$$\xi_{\min}^{\text{SA}} / \xi_{\min}^{\text{LMS}} = \eta_{\min}^{\text{SA}} / \eta_{\min}^{\text{LMS}} = \sqrt{\pi/2}. \quad (6.8)$$

Thus, ξ_{\min}^{SA} and η_{\min}^{SA} exceed ξ_{\min}^{LMS} and η_{\min}^{LMS} , respectively, by about 1 dB. Then, for Gaussian b_k and \underline{x}_k , the choice of SA versus LMS algorithm depends on how a 1-dB performance difference improvement compares to the difference in the implementation complexity of the two algorithms.

VII. SIMULATION RESULTS

The simulations are done for a case of adaptive identification of a randomly time-varying plant described by (2.6) where c ,

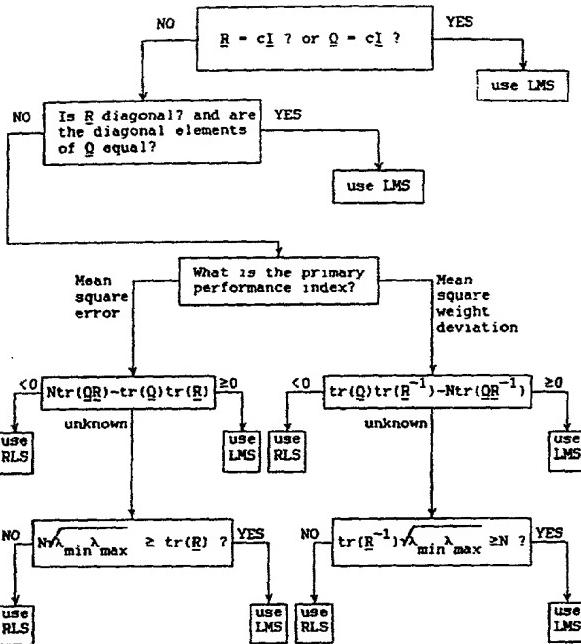


Fig. 1. Choice between the RLS and the LMS algorithms.

is the plant-parameter vector, b_k is the plant noise, and \underline{x}_k is the observation vector given by

$$\underline{x}_k^T = (x_k, x_{k-1}, \dots, x_{k-N+1}) \quad (7.1)$$

with x_k being the plant input. The sequences $\{b_k\}$ and $\{x_k\}$ are mutually independent, zero mean, and Gaussian. The sequence $\{b_k\}$ is white. The sequence $\{x_k\}$ is a stationary autoregressive sequence given by

$$x_k = \beta x_{k-1} + \sqrt{1-\beta^2} w_k, \quad 0 \leq \beta < 1 \quad (7.2)$$

where w_k is a zero mean unity variance white Gaussian sequence. The parameter β controls the degree of correlation of the sequence $\{x_k\}$; the greater β is, the stronger the correlation. The used model of variation of the plant-parameter vector \underline{c}_k is

$$\underline{c}_{k+1} = \underline{c}_k + \underline{d}_k, \quad \underline{d}_k \triangleq (w_{1,k}, w_{2,k}, \dots, w_{N,k})^T \quad (7.3)$$

where $\{w_{1,k}\}, \{w_{2,k}\}, \dots$, and $\{w_{N,k}\}$ are zero mean white Gaussian sequences with the same variance σ_w^2 . The sequences $\{w_{1,k}\}, \{w_{2,k}\}, \dots$, and $\{w_{N,k}\}$ are independent of the sequences $\{w_k\}$ and $\{b_k\}$.

The direct way of evaluating ξ and η is to find the averages of $e_k^2 - \sigma_b^2$ and $\|\underline{v}_k\|^2$ over a large number of independent runs for a fixed value of k in the steady state. An alternative simpler way, that can be used under the assumption that \underline{v}_k is ergodic, is to evaluate ξ and η from one run by averaging over a large number of iterations in the steady state. Since we have a large

TABLE II
ANALYTICAL AND SIMULATION RESULTS FOR
 $N = 2, \sigma_b = 0.2, \beta = 0.75, \sigma_w = 0.01$

CASE 1		CASE 2	
Analytical	Simulations	Analytical	Simulations
$\frac{\text{RLS}}{\text{LMS}} \frac{\xi_{\min}}{\xi_{\min}}$	0.5	0.575	1.325
$\frac{\text{SA}}{\text{LMS}} \frac{\xi_{\min}}{\xi_{\min}}$	1.25	1.075	1.25
$\frac{\text{RLS}}{\text{LMS}} \frac{\eta_{\min}}{\eta_{\min}}$	0.8	0.86	2
$\frac{\text{SA}}{\text{LMS}} \frac{\eta_{\min}}{\eta_{\min}}$	1.25	1.15	1.25

TABLE III
ANALYTICAL AND SIMULATION RESULTS FOR
 $N = 5, \sigma_b = 0.5, \beta = 0.75, \sigma_w = 0.005$

CASE 1		CASE 2	
Analytical	Simulations	Analytical	Simulations
$\frac{\text{RLS}}{\text{LMS}} \frac{\xi_{\min}}{\xi_{\min}}$	0.512	0.68	1.8
$\frac{\text{SA}}{\text{LMS}} \frac{\xi_{\min}}{\xi_{\min}}$	1.25	1.43	1.25
$\frac{\text{RLS}}{\text{LMS}} \frac{\eta_{\min}}{\eta_{\min}}$	0.73	0.81	3.13
$\frac{\text{SA}}{\text{LMS}} \frac{\eta_{\min}}{\eta_{\min}}$	1.25	1.18	1.25

Simulations have been carried out at several degrees of correlation between $w_{1,k}, w_{2,k}, \dots$, and $w_{N,k}$. Here, we give the results of the two following extreme cases:

Case 1: $w_{j+1,k} = -w_{j,k}$, $j = 1, 2, \dots, N-1$, for all k .

Case 2: $w_{j+1,k} = w_{j,k}$, $j = 1, 2, \dots, N-1$, for all k .

Table II shows analytical and simulation results of a 2-D case ($N = 2$) while Table III shows the results of a 5-D case. In Table II, $\sigma_b = 0.2, \sigma_w = 0.01$, while in Table III, $\sigma_b = 0.5$ and $\sigma_w = 0.005$. In both tables, $\beta = 0.75$. As seen by Tables II and III, both analytical and simulation results agree

VIII. CONCLUSIONS

The tracking capabilities of the RLS, LMS, and sign algorithms for adaptive FIR filters are compared in the case when the observation vector, \underline{x}_k , is stationary while the optimal filtering vector, \underline{c}_k , is randomly time-varying. The following conclusions are drawn from the paper:

- (1) The steady-state excess mean-square estimation error, ξ , does not depend on the eigenvalue spread of the covariance matrix $\underline{R} \triangleq E(\underline{x}_k \underline{x}_k^T)$ in the cases of the LMS algorithm and the sign algorithm, while it does in the case of the RLS algorithm.
- (2) For the three algorithms, the steady-state mean-square deviation η between the weight vector of the adaptive filter and \underline{c}_k depends on the eigenvalue spread of \underline{R} .
- (3) For the three algorithms, the value of the adaptation parameter that minimizes ξ is different from the one that minimizes η .
- (4) The ratio of the minimum value of ξ attained by the RLS algorithm to the one attained by the LMS algorithm is equal to $\sqrt{N \text{tr}(\underline{Q}\underline{R}) / (\text{tr}(\underline{R}) \text{tr}(\underline{Q}))}$, with N being the number of adaptive filter weights and \underline{Q} being the covariance matrix of the optimal vector increment $\underline{c}_{k+1} - \underline{c}_k$.
- (5) The ratio of the minimum value of η attained by the RLS algorithm to the one attained by the LMS algorithm is equal to $\sqrt{\text{tr}(\underline{Q}) \text{tr}(\underline{R}^{-1}) / (N \text{tr}(\underline{R}^{-1}))}$.
- (6) A sufficient condition for the tracking of the RLS algorithm to be worse (resp. better) than that of the LMS algorithm is that the optimal vector increment $\underline{c}_{k+1} - \underline{c}_k$ be biased, in mean-square sense, toward the directions of the eigenvectors of \underline{R} associated with the large (resp. small) eigenvalues.
- (7) The tracking capabilities of the RLS algorithm are the same as the ones of the LMS algorithm in any one of the three following cases: (1) if \underline{R} has equal eigenvalues, (2) if \underline{Q} has equal eigenvalues, or (3) if \underline{R} is diagonal and the diagonal elements of \underline{Q} are equal.
- (8) For Gaussian signals, the tracking performance of the sign algorithm is 1 dB lower than that of the LMS algorithm, independent of \underline{R} and \underline{Q} .

APPENDIX PROOF OF (6.4) AND (6.5)

With λ_i and \underline{u}_i being as defined in Section VI, one has

$$\underline{R} = \sum_{i=1}^N \lambda_i \underline{u}_i \underline{u}_i^T. \quad (I.1)$$

Since $\text{tr}(\underline{Q}\underline{R}) = E(\underline{d}_k^T \underline{R} \underline{d}_k)$, then (I.1) and (5.3) yield

$$\text{tr}(\underline{Q}\underline{R}) = \sum_{i=1}^N \lambda_i \gamma_i. \quad (I.2)$$

Equations (I.1) and (6.3), respectively, imply that $\text{tr}(\underline{R}) = \sum_{i=1}^N \lambda_i$, and $\text{tr}(\underline{Q}) = \sum_{i=1}^N \gamma_i$. Then, due to (I.2) one has

$$\begin{aligned} N \text{tr}(\underline{Q}\underline{R}) - \text{tr}(\underline{Q}) \text{tr}(\underline{R}) &= N \sum_{i=1}^N \lambda_i \gamma_i - \sum_{i=1}^N \sum_{j=1}^N \lambda_i \gamma_j \\ &= \sum_{i=1}^N \sum_{j=1}^N z_{i,j} \end{aligned} \quad (I.3)$$

where

$$z_{i,j} = \lambda_i \gamma_j - \lambda_j \gamma_i. \quad (I.4)$$

One has

$$\sum_{i=1}^N \sum_{j=1}^N z_{i,j} = \sum_{i=1}^N z_{i,i} + \sum_{i=1}^{N-1} \sum_{j=i+1}^N (z_{i,j} + z_{j,i}). \quad (I.5)$$

Due to (I.4), $z_{i,i} = 0$, $z_{i,j} + z_{j,i} = (\lambda_i - \lambda_j)(\gamma_i - \gamma_j)$, and then (I.3) and (I.5) yield (6.4).

To prove (6.5), we use (I.1) and (6.3) to obtain

$$\text{tr}(\underline{Q}\underline{R}^{-1}) = \sum_{i=1}^N \lambda_i^{-1} \gamma_i \quad (I.6)$$

and then

$$\begin{aligned} N \text{tr}(\underline{Q}\underline{R}^{-1}) - \text{tr}(\underline{Q}) \text{tr}(\underline{R}^{-1}) &= N \sum_{i=1}^N \lambda_i^{-1} \gamma_i - \sum_{i=1}^N \sum_{j=1}^N \lambda_i^{-1} \gamma_j. \end{aligned} \quad (I.7)$$

Continuing in a procedure similar to that given in (I.3)–(I.5), one can easily show that (I.7) implies (6.5).

REFERENCES

- [1] F. Ling and J. G. Proakis, "Nonstationary learning characteristics of least squares adaptive estimation algorithms," in *Proc. IEEE ICASSP 1984*, pp. 3.7.1–3.7.4.
- [2] E. Eleftheriou and D. D. Falconer, "Tracking properties and steady state performance of RLS adaptive filtering algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, no. 5, pp. 1097–1110, Oct. 1986.
- [3] O. Macchi, N. J. Bershad, and M. Mboup, "Steady-state superiority of LMS over LS for time-varying line enhancer in noisy environment," in *IEE Proc. Part F—Radar and Signal Processing*, vol. 138, no. 4, Aug. 1991, pp. 354–360.
- [4] E. Eweda and O. Macchi, "Convergence of the RLS and LMS adaptive filters," *IEEE Trans. Circuits Syst.*, vol. CAS-34, no. 7, pp. 799–803, July 1987.
- [5] O. Macchi and E. Eweda, "Compared speed and accuracy of the RLS and LMS algorithms with constant forgetting factors," *APII*, vol. 22, pp. 255–267, Mar. 1988.
- [6] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, Jr., "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, pp. 1151–1162, Aug. 1976.
- [7] S. H. Cho and V. J. Mathews, "Tracking analysis of the sign algorithm in nonstationary environments," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, no. 12, pp. 2046–2057, Dec. 1990.
- [8] N. A. M. Verhoeckx and T. A. C. M. Claasen, "Some considerations on the design of adaptive digital filters equipped with the sign algorithm," *IEEE Trans. Commun.*, vol. COM-32, no. 3, pp. 258–266, Mar. 1984.
- [9] E. Eweda, "Optimum step size of sign algorithm for nonstationary adaptive filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, no. 11, pp. 1897–1901, Nov. 1990.

- [10] R. R. Bitmead and B. D. Anderson, "Performance of adaptive estimation algorithms in dependent random environments," *IEEE Trans. Automat. Contr.*, vol. AC-25, no. 4, pp. 788-794, 1980.
- [11] B. Widrow *et al.*, "Adaptive noise cancelling: Principles and applications," *Proc. IEEE*, vol. 63, pp. 1692-1716, Dec. 1975.
- [12] O. Macchi, "Optimization in adaptive identification for time-varying filters," *IEEE Trans. Automat. Contr.*, vol. AC-31, no. 3, pp. 283-287, Mar. 1986.
- [13] E. Eweda, "Analysis and design of a signed regressor LMS algorithm for stationary and nonstationary adaptive filtering with correlated Gaussian data," *IEEE Trans. Circuits Syst.*, vol. 37, no. 11, pp. 1367-1374, Nov. 1990.
- [14] L. Ljung and S. Gunnarsson, "Adaptation and tracking in system identification—A survey," *Automatica*, vol. 26, no. 1, pp. 7-21, Jan. 1990.
- [15] S. T. Alexander, *Adaptive Signal Processing: Theory and Applications*. New York: Springer-Verlag, 1986.



Eweda Eweda was born in Kena, Egypt, on October 13, 1950. He received the B.Sc. and M.Sc. degrees from the Military Technical College, Cairo, Egypt in 1973 and 1977, respectively, and the Ph.D. and Doctor of Science degrees from Paris XI University, Paris, France, in 1980 and 1983, respectively.

Currently, he is a Professor in the Department of Electrical Engineering at the Military Technical College, Cairo, Egypt. He is the author of many journal papers on adaptive filtering with correlated quantized, and nonstationary data. One of his papers

was elected for inclusion in the IEEE Press reprint book, "Adaptive Signal Processing," in 1987. His current scientific interests include aspects of adaptive signal processing, data transmission, and communication theory.

Dr. Eweda has received a highly outstanding Egyptian scientific award, The State Award of Engineering Sciences.

Unitary ESPRIT: How to Obtain Increased Estimation Accuracy with a Reduced Computational Burden

Martin Haardt, *Student Member, IEEE* and Josef A. Nossek, *Fellow, IEEE*



Abstract—ESPRIT is a high-resolution signal parameter estimation technique based on the translational invariance structure of a sensor array. Previous ESPRIT algorithms do not use the fact that the operator representing the phase delays between the two subarrays is unitary. Here, we present a simple and efficient method to constrain the estimated phase factors to the unit circle if centro-symmetric array configurations are used. Unitary ESPRIT, the resulting closed-form algorithm, has an ESPRIT-like structure except for the fact that it is formulated in terms of real-valued computations throughout. Since the dimension of the matrices is not increased, this completely real-valued algorithm achieves a substantial reduction of the computational complexity. Furthermore, Unitary ESPRIT incorporates forward-backward averaging, leading to an improved performance compared to the standard ESPRIT algorithm, especially for correlated source signals. Like standard ESPRIT, Unitary ESPRIT offers an inexpensive possibility to reconstruct the impinging wavefronts (*signal copy*). These signal estimates are more accurate, since Unitary ESPRIT improves the underlying signal subspace estimates. Simulations confirm that, even for uncorrelated signals, the standard ESPRIT algorithm needs twice the number of snapshots to achieve a precision comparable to that of Unitary ESPRIT. Thus, Unitary ESPRIT provides increased estimation accuracy with a reduced computational burden.

structure, Unitary ESPRIT provides a very simple and efficient solution to this task.

Although Unitary ESPRIT effectively doubles the number of data samples, the computational complexity is reduced by transforming the required rank-revealing factorizations of complex matrices into decompositions of *real-valued* matrices of the *same size*. Thus, we obtain increased estimation accuracy with a reduced computational load. This reduction can be achieved by constructing invertible transformations that map centro-Hermitian matrices to real matrices. These transformations have been introduced in Lee's pioneering work on centro-Hermitian matrices [10]. More than a decade later, her results were used to transform the complex covariance matrix of a uniform linear array (ULA) into a real matrix of the same size [8] to reduce the computational load of adaptive beamforming schemes [9]. In this paper, we use more general centro-symmetric array configurations that have been receiving increased attention lately [22]. We derive an efficient square root version of Unitary ESPRIT that only requires real-valued computations from start to finish, by operating directly on the data instead of "squaring" it to obtain sample covariance matrices. It is well known that benefits result from smaller matrix conditioning numbers [15]. With infinite precision, both strategies would be the same, whether eigendecompositions or singular value decompositions (SVD's) are used. Finite precision arithmetic, however, is employed in practical applications. Therefore, numerical issues like round-off error and overflow are potential problems to be aware of when covariance matrices are estimated. In addition to this square root approach, we also describe an alternative real-valued covariance approach, cf. Remark 1, which turns out to be more efficient than the one proposed in [8] and [24].

In the presence of additive noise, the computation of an optimal signal subspace estimate requires an SVD or an eigenvalue decomposition (EVD), which is computationally expensive, since $\mathcal{O}(M^3)$ operations are necessary to update the SVD or EVD if a new sample vector of dimension M arrives. Therefore, a number of alternative decompositions have been proposed to estimate the signal subspace in a computationally more efficient way. Examples include the rank-revealing QR decomposition [1], [2], the rank-revealing URV decomposition [11], [18], or a new Schur-type method for subspace estimation [3], [9]. These approximation techniques are computationally more efficient and well suited for a parallel (systolic) implementation, but they involve a certain

THE recovery of signal parameters from noisy observations is a fundamental problem in (real-time) array signal processing. Due to their simplicity and high-resolution capability, ESPRIT-like subspace estimation schemes have been attracting considerable attention. Their parameter estimates are obtained by exploiting the *rotational* invariance structure of the signal subspace, induced by the *translational* invariance structure of the associated sensor array. This can be achieved without computation or search of any spectral measure [15], [17]. Unitary ESPRIT achieves even more accurate results than previous ESPRIT techniques by taking advantage of the unit magnitude property of the phase factors that represent the phase delays between the two subarrays [4]. It has been shown in [12] that constraining the phase factors to the unit circle can also give some improvement for correlated sources. For centro-symmetric sensor arrays with a translational invariance

Manuscript received June 8, 1994; revised August 30, 1994. The associate editor coordinating the review of this paper and approving it for publication was Prof. Keshab Parhi.

The authors are with the Institute of Network Theory and Circuit Design, Technical University of Munich, Munich, Germany.

IEEE Log Number 9410301.

loss of accuracy that can be compensated by combining them with Unitary ESPRIT, yielding not only improved estimation accuracy, but also completely real-valued algorithms. In [5], it is shown how Unitary Schur ESPRIT dramatically improves the performance of the new Schur-type method, an adaptive subspace estimation scheme with a computational structure and complexity similar to that of a QR decomposition, except for the fact that plane and hyperbolic rotations are used. In this case, the required rank-decision, i.e., an estimate of the number of signals, is automatic, and updating as well as downdating are straightforward. The fully real-valued Unitary ESPRIT concept can also be extended to spatially smoothed forward-backward estimation schemes [6], [13], [14] and is applicable to many other subspace estimation techniques (see [20] for an excellent overview). The results are comparable to the advantages obtained by operating in beampspace [24] without the necessity of converting the data from element space to beampspace.

This paper is organized as follows. It starts with a review of the definition and basic properties of centro-Hermitian matrices. These properties will be used to derive the real-valued implementation of Unitary ESPRIT. A brief review of the standard ESPRIT algorithm is given in chapter III. It can be seen as a generalization of the matrix pencil method [7]. Chapter IV introduces the Unitary ESPRIT concept for centro-symmetric array structures. In Section IV-B we show how all three required rank-revealing factorizations can be transformed into decompositions of *real-valued* matrices of the same size yielding a completely real algorithm. A new reliability test, which is a substantial improvement of current high-resolution array signal processing and spectral estimation techniques, is presented in Section IV-C. Further simplifications of the algorithm are derived in Section IV-D, before a summary of Unitary ESPRIT concludes the chapter (Section IV-E). Finally, computer simulations compare the performance of Unitary ESPRIT with that of the well-known standard ESPRIT algorithm (Section V).

II. CENTRO-HERMITIAN MATRICES

First of all, let us introduce our notation and review the definition and the basic properties of centro-Hermitian matrices that have been derived by Lee [10]. Throughout this paper, column vectors and matrices are denoted by lower case and upper case boldfaced letters, respectively. Π_p is the $p \times p$ exchange matrix with ones on its antidiagonal and zeros elsewhere

$$\Pi_p = \begin{bmatrix} & & & 1 \\ & & 1 & \\ & \ddots & & \\ 1 & & & \end{bmatrix} \in \mathbb{R}^{p \times p}.$$

Since Π_p is a symmetric permutation matrix, it is involutorial, i.e., $\Pi_p^2 = I_p$. With this notation, we can define centro-Hermitian matrices in analogy to centro-symmetric matrices.

Definition 1 A complex matrix $M \in \mathbb{C}^{p \times q}$ is called centro-Hermitian if

where the overbar denotes complex conjugation without transposition.

Centro-Hermitian matrices of size $p \times q$ form a $p \times q$ -dimensional linear space over \mathbb{R} [10]. To show how centro-Hermitian matrices can be mapped to matrices with real entries, Lee defines left Π -real matrices in the following fashion.

Definition 2 [10]: Matrices $Q \in \mathbb{C}^{p \times q}$ satisfying

$$\Pi_p Q = Q$$

are *left Π -real*.

The unitary matrices

$$Q_{2n} = \frac{1}{\sqrt{2}} \begin{bmatrix} I_n & jI_n \\ \Pi_n & -j\Pi_n \end{bmatrix} \quad (3)$$

$$Q_{2n+1} = \frac{1}{\sqrt{2}} \begin{bmatrix} I_n & 0 & jI_n \\ 0^T & \sqrt{2} & 0^T \\ \Pi_n & 0 & -j\Pi_n \end{bmatrix} \quad (4)$$

for example, are left Π -real of even and odd order, respectively. More left Π -real matrices can be obtained by post-multiplying a left Π -real matrix Q by an arbitrary real matrix R , i.e., every matrix QR is left Π -real. Now, we are in a position to state Lee's main result, which establishes an automorphism between centro-Hermitian and real matrices.

Theorem 1 [10]: Let T_p and U_q denote arbitrary nonsingular left Π -real matrices of size $p \times p$ and $q \times q$, respectively. Then, the bijective mapping

$$\varphi: M \mapsto T_p^{-1} M U_q$$

maps the set of all $p \times q$ centro-Hermitian matrices onto $\mathbb{R}^{p \times q}$, the set of all real matrices of the same size.

This theorem can, for instance, be used to calculate the singular value decomposition (SVD) of a centro-Hermitian matrix $M \in \mathbb{C}^{p \times q}$.

Corollary 1: Let M be centro-Hermitian, and assume that the SVD of $\varphi_Q(M) = Q_p^H M Q_q \in \mathbb{R}^{p \times q}$ is given by $\varphi_Q(M) = U_\varphi \Sigma_\varphi V_\varphi^H$, where the matrices Q_p and Q_q are unitary as well as left Π -real. Then, an SVD of M is obtained as

$$M = (Q_p U_\varphi) \Sigma_\varphi (V_\varphi^H Q_q^H) \quad \text{see end} \quad (5)$$

where the left and right singular vectors of M are left Π -real.

Proof: The first part follows from the unitary nature of Q_p and Q_q , the second from the fact that the singular vectors of a real matrix are real. ■

For future reference, we consider an efficient computation of a particular transformation $T(\cdot)$. It transforms an arbitrary complex matrix $G \in \mathbb{C}^{p \times q}$ into a real $p \times 2q$ matrix, denoted by $T(G)$. Notice that for every matrix G , the matrix

$$(G \quad \Pi_p G \Pi_q) \in \mathbb{C}^{p \times 2q}$$

is centro-Hermitian. Thus, the matrix

$$T(G) \stackrel{\text{def}}{=} \varphi_Q([G \quad \Pi_p G \Pi_q]) \\ = Q_p^H [G \quad \Pi_p G \Pi_q] Q_q$$

$$\stackrel{\text{unitary}}{=} Q_p^H [G \quad \Pi_p G \Pi_q] Q_q \quad \text{mod } (6)$$

Recall that the SVD of a complex matrix is unique up to a unitary diagonal

is always real according to Theorem 1. Consider the case where the left H -real matrices Q_p and Q_{2q} are chosen according to (3) or (4). Furthermore, let G be partitioned as

$$G = \begin{bmatrix} G_1 \\ g^T \\ G_2 \end{bmatrix}$$

where the block matrices G_1 and G_2 should have the same size. Obviously, the row vector g^T must be dropped if p is even. Then, straightforward calculations show that the desired real-valued matrix (6) can be expressed as

$$\rightarrow T(G) = \begin{bmatrix} \operatorname{Re}\{G_1 + H\bar{G}_2\} & -\operatorname{Im}\{G_1 - H\bar{G}_2\} \\ \sqrt{2} \cdot \operatorname{Re}\{g^T\} & -\sqrt{2} \cdot \operatorname{Im}\{g^T\} \\ \operatorname{Im}\{G_1 + H\bar{G}_2\} & \operatorname{Re}\{G_1 - H\bar{G}_2\} \end{bmatrix}. \quad (7)$$

Here, $\operatorname{Re}\{\cdot\}$ and $\operatorname{Im}\{\cdot\}$ denote the real and the imaginary part, respectively. Once again, if p is odd, the center row of (7) should be dropped. Then, an efficient computation of $T(G) \in \mathbb{R}^{p \times 2q}$ from the complex matrix G only requires $p \times 2q$ real additions.

III STANDARD ESPRIT

A. Standard ESPRIT Scenario

Consider the standard ESPRIT scenario [15], [17], i.e., an M -element sensor array composed of m pairs of pairwise identical, but displaced sensors (doublets). Let Δ denote the distance between the two subarrays. Incident on both subarrays are d narrow-band noncoherent planar wavefronts

$$s_k(t_n) = u(t_n) e^{j(\omega_0 t_n + v(t_n))}, \quad 1 \leq k \leq d \leq m$$

with signal propagation velocity c and a common center frequency ω_0 . The d impinging signals are combined to a signal vector $s(t_n)$. Assume, for the moment, that the two subarrays do not share any elements, i.e., they do not overlap. Then, the total number of sensors equals $M = 2m$ and the uncorrupted signals received at the two subarrays have the following form:

$$x(t_n) = \begin{bmatrix} x_1(t_n) \\ x_2(t_n) \end{bmatrix} = \begin{bmatrix} A \\ A\Phi \end{bmatrix} s(t_n) = A_G s(t_n). \quad (8)$$

$A_G \in \mathbb{C}^{M \times d}$ and $A \in \mathbb{C}^{m \times d}$ are the steering matrices of the whole array configuration (global array steering matrix) and the first subarray, respectively. Notice that the k th columns of both array steering matrices depend on the direction of arrival (DOA) θ_k of the k th source relative to the displacement between the two subarrays.² Furthermore

$$\Phi = \operatorname{diag}\{\phi_k\}_{k=1}^d \in \mathbb{C}^{d \times d}$$

is a diagonal matrix of the phase delays between the sensor doublets for the d wavefronts. Its diagonal elements, the phase factors ϕ_k , are given by

$$\phi_k = e^{-j\omega_0 \Delta \sin \theta_k / c} \stackrel{\text{def}}{=} e^{j\mu_k}, \quad 1 \leq k \leq d. \quad (9)$$

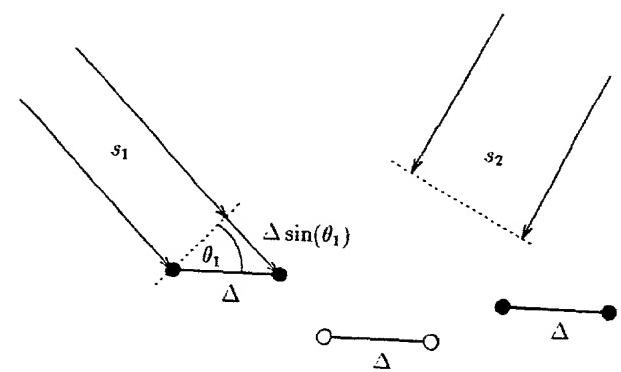


Fig. 1 Planar array composed of $m = 3$ pairwise identical but displaced sensors (doublets).

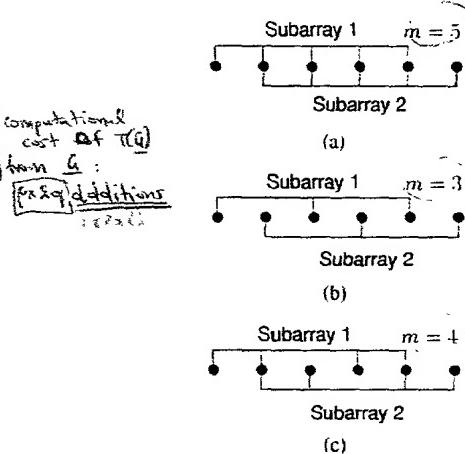


Fig. 2. Three different subarray choices for a uniform linear array (ULA) of $M = 6$ identical sensors. (a) Maximum overlap ($m = 5$); (b) interleaved ($m = 3$); (c) mixed ($m = 4$).

B. More Structured Array Geometries

Recall that every row of A_G corresponds to an element of the sensor array. In the case of overlapping subarrays, a particular subarray configuration is described by selection matrices that choose m elements of $x(t_n) \in \mathbb{C}^M$, where $m < M$ is the number of elements in each subarray. Let J_1 and J_2 be $(m \times M)$ selection matrices that assign elements of $x(t_n)$ to the subarrays one and two, respectively. Fig. 2, for example, displays three different subarray choices for a uniform linear array (ULA) of $M = 6$ identical sensors.

In general, the two selection matrices are chosen to be centro-symmetric with respect to one another, i.e.

$$J_2 = \Pi_m J_1 \Pi_M \quad (10)$$

a property that plays a key role in the derivation of the fully real implementation of Unitary ESPRIT, cf. Section IV-B. Therefore, the combined selection matrix

$$J \stackrel{\text{def}}{=} \begin{bmatrix} J_1 \\ J_2 \end{bmatrix} \in \mathbb{R}^{2m \times M}$$

is centro-Hermitian, i.e. $\Pi_{2m} J \Pi_M = J$.

C General ESPRIT Principle

By collecting $N >$ snapshots from each sensor $1 \leq n \leq N$,

are formed, obeying

$$\begin{aligned} JX &= J[x(t_1) \quad x(t_2) \quad \cdots \quad x(t_N)] \\ &= \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} A \\ A\Phi \end{bmatrix} S. \end{aligned} \quad (11)$$

It is easy to see that every row in X corresponds to an element of the sensor array. Equation (11) implies that X_1 , X_2 , and X are rank-deficient, namely $\text{rank } X_1 = \text{rank } X_2 = \text{rank } X = d$. Thus, the d columns of

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} P_{\text{col}} = \begin{bmatrix} A \\ A\Phi \end{bmatrix} SP_{\text{col}}$$

form a basis for the column space of X if

$$SP_{\text{col}} \in \text{GL}(d). \quad (12)$$

Here, $\text{GL}(d) \subset \mathbb{C}^{d \times d}$ denotes the general linear group of all nonsingular matrices of dimension $d \times d$. Observe that the column space or range of JX , range $JX \subset \mathbb{C}^{2m}$, is usually called *signal subspace*. In the same way, the d rows of

$$\begin{aligned} \Gamma_1 \quad \Gamma_2 &= P_{\text{row}}[C_1 \quad C_2] \\ &= P_{\text{row}}A[SP_{\text{col}} \quad \Phi SP_{\text{col}}] \end{aligned}$$

form a basis for the row space of $[C_1 \quad C_2]$ if

$$P_{\text{row}}A \in \text{GL}(d). \quad (13)$$

Therefore, the rank-reducing numbers of the matrix pencil

$$\Gamma_2 - \lambda \Gamma_1 = P_{\text{row}}A(\Phi - \lambda I_d)SP_{\text{col}},$$

are the diagonal elements of Φ (phase factors) and can be calculated as the generalized eigenvalues of the matrix pair (Γ_2, Γ_1) .

Due to these observations, the ESPRIT algorithm reduces to choosing the appropriate compression matrices that define the required bases. In the absence of noise (the case discussed so far), any matrices (P_{col}) and (P_{row}) satisfying (12) and (13) will do the job. With noisy measurements, however, we are faced with the problem of estimating the signal subspace and its dimension.

D. SVD-Based Subspace Estimate

The most robust way to estimate the required bases is to compute the singular value decomposition (SVD) of

$$\tilde{X} = [U_s \quad U_o] \begin{bmatrix} \Sigma_s & \\ & \Sigma_o \end{bmatrix} \begin{bmatrix} V_s^H \\ V_o^H \end{bmatrix} \quad (14)$$

where \tilde{X} denotes the measurement matrix X corrupted by additive, spatially uncorrelated³ noise, Σ contains its d dominant singular values, and the unitary matrices U and V are partitioned accordingly. Then, the best rank d approximation of \tilde{X} in the Frobenius-norm is given by $\tilde{X} = U_s \Sigma_s V_s^H$. In other words, our low rank estimate of X is the matrix \tilde{X} satisfying

$$\|\tilde{X} - X\|_F = \min_{\text{rank } Y \leq d} \|\tilde{X} - Y\|_F. \quad (15)$$

³If the spatial covariance matrix of the additive noise is known up to a scalar factor, the SVD can be replaced by the generalized or quotient SVD (GSVD) as described in [17].

A basis for the estimated signal subspace is determined from the d dominant left singular vectors according to

$$JU_s = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$$

Then, a unitary basis for the row space of $[C_1 \quad C_2]$ can also be obtained by computing its SVD (total least squares approach). However, it is less expensive to use $P_{\text{row}} = C_1^H$, which corresponds to the standard *least squares* solution of the overdetermined set of equations

$$C_1 \Psi \approx C_2 \quad (16)$$

followed by an eigendecomposition of $\Psi = \Gamma_1^{-1} \Gamma_2$:

IV. UNITARY ESPRIT

A. Multiple Invariance Structure

Unitary ESPRIT is applicable to *centro-symmetric array configurations*. A sensor array is called centro-symmetric [22] if its element locations are symmetric with respect to the centroid and the complex characteristics of paired elements are the same. Their global array steering matrix A_G , therefore, satisfies

$$\Pi_M \bar{A}_G = A_G \Lambda_G \quad (17)$$

for some unitary diagonal matrix $\Lambda_G \in \mathbb{C}^{d \times d}$. Notice that the matrix $A_G \Lambda_G^{1/2}$ is left Π -real.

Uniform linear arrays, for example, the most common arrays used in practice, are centro-symmetric. It is well known that the analogy between array signal processing and time series analysis (harmonic retrieval) can be obtained through uniform linear arrays (ULA's) by interpreting them as uniform sampling of a time series [16].

The centro-symmetry of the global sensor array A_G and (10) imply that the steering matrices of both subarrays are also centro-symmetric, i.e.

$$\Pi_M \bar{A} = A \Lambda \quad \text{with } \Lambda = \Phi \Lambda_G.$$

Without additive noise, the Unitary ESPRIT data matrix

$$Z \stackrel{\text{def}}{=} [X \quad \Pi_M \bar{X}]$$

admits the factorization

$$\begin{aligned} JZ &= \begin{bmatrix} X_1 & \Pi_M \bar{X}_2 \\ X_2 & \Pi_M \bar{X}_1 \end{bmatrix} \\ &= \begin{bmatrix} A \\ A\Phi \end{bmatrix} [S \quad \Phi^{-1} \bar{A} \bar{S}] \end{aligned} \quad (18)$$

which is easily seen by using the centro-symmetry of the subarrays and the unitary nature of Φ . Thus, Z is also rank-deficient, namely $\text{rank } Z = d$. Equations (11) and (18) show that Unitary ESPRIT essentially doubles the number of available measurements from N to $2N$. Increased estimation accuracy can, therefore, be achieved by replacing the measurement matrix $X \in \mathbb{C}^{M \times N}$ of the standard ESPRIT formulation (11) by $Z \in \mathbb{C}^{M \times 2N}$, which corresponds to forward-backward averaging of the data.

B. Real Implementation

Due to the special algebraic structure of the noise-corrupted data matrix \tilde{Z} and the structure of the subsequent total least squares (TLS) problem, the computational complexity of Unitary ESPRIT can be reduced significantly. This is achieved by transforming the three (complex-valued) rank-revealing factorizations,

- the subspace estimation step
- the subsequent total least squares problem
- the final eigenvalue decomposition (EVD)

into factorizations of *real-valued* matrices of the *same size*. Thus, real-valued computations can be maintained for all steps of the Unitary ESPRIT algorithm. The following three propositions derive the required transformations by taking advantage of the mapping between centro-Hermitian and real matrices, cf. Section II. In Remark 3, we also show how the real-valued total least squares problem can be replaced by a real-valued least squares (LS) problem.

Proposition 1—Signal Subspace Estimation: The principal subspace of $\tilde{Z} \in \mathbb{C}^{M \times 2N}$ (and, therefore, also the principal subspace of $J\tilde{Z}$) can be obtained through a rank-revealing factorization of the real matrix $T(\tilde{X}) \in \mathbb{R}^{M \times 2N}$, where the transformation $T(\cdot)$ is defined in (6). Then, the complex matrices C_1 and C_2 , spanning the estimated signal subspace, obey

$$C_2 = \Pi_m \bar{C}_1. \quad (19)$$

Proof. By post-multiplying the noise-corrupted matrix \tilde{Z} with a unitary permutation matrix we obtain a centro-Hermitian matrix in the following fashion:

$$\tilde{Z}_{CH} = \tilde{Z} \begin{bmatrix} I_N & \\ & \Pi_N \end{bmatrix} = [\tilde{X} \quad \Pi_M \bar{\tilde{X}} \Pi_N]. \quad (20)$$

According to Corollary 1, a rank-revealing factorization of \tilde{Z}_{CH} can, thus, be obtained through an SVD of the real matrix

$$\varphi_Q(\tilde{Z}_{CH}) = Q_M^H \tilde{Z}_{CH} Q_{2N} = T(\tilde{X}) \quad (21)$$

which proves the first part of this proposition. Let the d dominant left singular vectors of $\varphi_Q(\tilde{Z}_{CH})$ be denoted by $E_s \in \mathbb{R}^{M \times d}$. Then, the d dominant left singular vectors of \tilde{Z}_{CH} as well as \tilde{Z} are given by $Q_M E_s$. Therefore, the matrix

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} J_1 \\ J_2 \end{bmatrix} Q_M E_s$$

provides a basis for the estimated signal subspace. With (10) and the left Π -realness of Q_M we, finally, get the desired result:

$$\begin{aligned} \Pi_m \bar{C}_1 &= \Pi_m J_1 \bar{Q}_M E_s \\ &= \Pi_m \Pi_m J_2 \Pi_M \bar{Q}_M E_s \\ &= J_2 Q_M E_s = C_2. \end{aligned}$$

Proposition 2—Total Least Squares Problem: The complex-valued SVD of size $m \times 2d$ that solves the total least squares (TLS) problem $C_1 \Psi \approx C_2$, which is associated with Unitary ESPRIT, can be transformed into an SVD of the real matrix $T(C_1) \in \mathbb{R}^{m \times 2d}$, where the transformation $T(\cdot)$ is defined in (6). Moreover, the eigenvalues ϕ_k of the resulting TLS solution $\Psi_{TLS} \in \mathbb{C}^{d \times d}$ will be symmetric with respect to the unit circle, i.e., there are indices

$$k, l \in \{1, 2, \dots, d\} \text{ such that } \phi_k = \frac{1}{\phi_l}. \quad (22)$$

Proof. The multidimensional TLS problem $C_1 \Psi \approx C_2$ can be solved through an SVD of

$$[C_1 \quad C_2] = [U_1 \quad U_2] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_{11}^H & V_{21}^H \\ V_{12}^H & V_{22}^H \end{bmatrix}. \quad (22)$$

Then, the TLS solution is obtained from the d right singular vectors corresponding to the d smallest singular values according to

$$\Psi_{TLS} = -V_{12} V_{22}^{-1} \quad (23)$$

where we have assumed that $V_{22} \in \text{GL}(d)$, i.e., the TLS solution is unique. For the singular case, the reader is referred to [21].

Thus, the TLS problem associated with Unitary ESPRIT can be solved through an SVD of

$$[C_1 \quad C_2] \stackrel{(19)}{=} [C_1 \quad \Pi_m \bar{C}_1].$$

Notice that this matrix has the same structure as

$$\tilde{Z} = [\tilde{X} \quad \Pi_M \bar{\tilde{X}}].$$

Using, therefore, the same reasoning as in (20) and (21), the TLS problem is solved by computing an SVD of the real matrix

$$T(C_1) = Q_m^H [C_1 \quad \Pi_m \bar{C}_1] \begin{bmatrix} I_d & \\ & \Pi_d \end{bmatrix} Q_{2d}. \quad (24)$$

Its right singular vectors will be denoted by

$$W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \in \mathbb{R}^{2d \times 2d}.$$

Then, the right singular vectors of $[C_1 \quad C_2]$ are determined from

$$V^H = \begin{bmatrix} V_{11}^H & V_{21}^H \\ V_{12}^H & V_{22}^H \end{bmatrix} = W^H Q_{2d}^H \begin{bmatrix} I_d & \\ & \Pi_d \end{bmatrix} \quad (25)$$

and Ψ_{TLS} is obtained from (23). Since the matrix $Q_{2d} W$ is left Π -real, it can be written as

$$Q_{2d} W = \begin{bmatrix} V_1 \\ \Pi_d \bar{V}_1 \end{bmatrix}$$

for some matrix $V_1 \in \mathbb{C}^{d \times 2d}$, cf. (2). With (25) we, therefore, conclude $V_{22} = \bar{V}_{12}$. Thus, if ϕ_l is an eigenvalue of the TLS solution $\Psi_{TLS} \in \text{GL}(d)$, $1/\phi_l$ is an eigenvalue of

$$\bar{\Psi}_{TLS}^{-1} = -V_{12} \bar{V}_{12}^{-1} = \Psi_{TLS}$$

Proposition 3 *Eigenvalue Decomposition:* The eigenvalues of the complex matrix Ψ_{TLS} can be determined from the eigenvalues of a real matrix of size $d \times d$ via the linear fractional transformation

$$f(x) = -\frac{x-j}{x+j} \quad (26)$$

Moreover, the eigenvectors of both matrices are identical.

Proof (a) Assume, for the moment, that the left \mathcal{H} -real matrix Q_{2d} is the one we have defined in (3). Then, (25) yields

$$\begin{aligned} V &= \begin{bmatrix} I_d & \\ & H_d \end{bmatrix} Q_{2d} W \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} I_n & jI_n \\ I_n & -jI_n \end{bmatrix} W. \end{aligned}$$

After partitioning V and W as before, we therefore conclude from (23)

$$\begin{aligned} \Psi_{\text{TLS}} &= -(W_{12} + jW_{22})(W_{12} - jW_{22})^{-1} \\ &= -((-W_{12}W_{22}^{-1}) - jI_d)((-W_{12}W_{22}^{-1}) + jI_d)^{-1} \\ &= f(\mathcal{T}_{\text{TLS}}) \quad \text{with } \mathcal{T}_{\text{TLS}} = -W_{12}W_{22}^{-1}. \end{aligned} \quad (27)$$

Here, $f(x)$ denotes the linear fractional transformation (26), which is analytic for $x \neq -j$. Let

$$\mathcal{T}_{\text{TLS}} = -W_{12}W_{22}^{-1} = T\Omega T^{-1} \quad (28)$$

be the eigenvalue decomposition (EVD) of the real matrix \mathcal{T}_{TLS} . It is a well-known result from function theory that the eigenvalues of Ψ_{TLS} can be obtained through the same linear fractional transformation, i.e.

$\Phi = f(\Omega)$ with $\Omega = \text{diag}\{\omega_k\}_{k=1}^d$ and $\omega_k \neq -j$

and the corresponding eigenvectors of \mathcal{T}_{TLS} and Ψ_{TLS} are identical.

(b) An arbitrary left \mathcal{H} -real matrix of dimension $2d \times 2d$ can, obviously, be written as

$$\tilde{Q}_{2d} = Q_{2d} R_{2d} \quad \text{where } R_{2d} \in \mathbb{R}^{2d \times 2d}.$$

After replacing the real matrix W by $\tilde{W} = R_{2d}W$, we invoke the same reasoning as above to prove this proposition for an arbitrary left \mathcal{H} -real transformation \tilde{Q}_{2d} . ■

Remark 1 *Covariance Approach:* Instead of the described square root (or direct data) approach based on a real-valued SVD, cf. Proposition 1, we can use a covariance approach based on a real-valued EVD to determine the signal subspace estimate. Then, $E_s \in \mathbb{R}^{M \times d}$ denotes the d principal eigenvectors of

$$T(\tilde{X})T(\tilde{X})^H \in \mathbb{R}^{M \times M}. \quad (29)$$

First, forming $T(\tilde{X})$ according to (7), followed by the computation of (29), is more efficient than the approach alternative suggested in [8] and [24]. There, it is proposed to compute the complex-valued sample covariance matrix $R_{XX} = \tilde{X}\tilde{X}^H \in \mathbb{C}^{M \times M}$ first. Then, E_s is determined from the EVD of $\text{Re}\{O_v^H R_{XX} O_v\}$, which is computationally more expensive.

C. Reliability Test

Proposition 2 states that the eigenvalues of Ψ_{TLS} , i.e., the phase factors estimated via Unitary ESPRIT, are symmetric with respect to the unit circle, since they satisfy (22). This observation gives rise to a new reliability test provided by Unitary ESPRIT without demanding additional computations. This reliability test is a substantial improvement of current high-resolution array signal processing and spectral estimation techniques since usually there is no easy way to determine how reliable the resulting estimates are. Unreliable estimation results might have been caused by a false estimate of the number of sources d or by the fact that there is no source signal at all (only noise).

Remark 2 *Eigenvalues with Unit Modulus:* Notice first that the eigenvalues ϕ_k that lie on the unit circle form a subset with nonzero measure in the class of all eigenvalues fulfilling (22), i.e., being symmetric with respect to the unit circle. Owing to this and the fact that Unitary ESPRIT produces consistent DOA estimates, asymptotically all the estimated phase factors ϕ_k will be on the unit circle.

If, however, the number of snapshots N is too small or if there is only noise present, the eigenvalues of Ψ_{TLS} might fail to satisfy

$$|\phi_k| = 1 \quad \forall k \quad (30)$$

which indicates that the subsequent estimates will be unreliable. Hence, no further computations should be carried out. Condition (30) implies that all eigenvalues ω_k of \mathcal{T}_{TLS} are real, cf. (26).⁴ Thus, if some of the ω_k occur in complex conjugate pairs, the Unitary ESPRIT reliability test has failed, and the algorithm has to be restarted with an increased window length N or more reliable measurements. If, conversely, all eigenvalues ω_k are real, i.e., the reliability test has been “passed,” all estimated phase factors ϕ_k are precisely on the unit circle.

D. Real-Valued Least Squares

Notice that the derivation of the Unitary ESPRIT reliability test is based on a total least squares solution of (16). Thus, the computation of \mathcal{T}_{TLS} requires an SVD (or another rank revealing factorization) of $T(C_1) \in \mathbb{R}^{m \times 2d}$. By computing the less expensive least squares instead of the total least squares solution of (16), we would, however, lose the benefits of the reliability test, since (22) would no longer be satisfied. Moreover, the complex-valued least squares problem (16) cannot be transformed into a real-valued problem of the same size. The following remark, however, sets up a different real-valued least squares problem, which can be solved instead.

Remark 3 *Least Squares Estimate:* After partitioning the real matrix of (24) according to

$$T(C_1) \stackrel{\text{def}}{=} [T_1 \ T_2] \quad \text{with } T_1, T_2 \in \mathbb{R}^{m \times d}$$

it is easy to see that \mathbf{T}_{TLS} is a TLS solution of the real-valued system of equations

$$\mathbf{T}_1 \mathbf{T} \approx \mathbf{T}_2. \quad (31)$$

To save computations, we can, therefore, solve (31) by computing its *least squares* solution \mathbf{T}_{LS} . Here, the Unitary ESPRIT reliability test is still applicable, since the resulting matrix \mathbf{T}_{LS} is always real. If the reliability test has been "passed", the estimated phase factors are on the unit circle.

The real-valued LS or TLS problem (31) can directly be obtained from \mathbf{E}_s by observing

$$\begin{aligned} \mathcal{T}(C_1) &= \mathbf{Q}_m^H [C_1 \ C_2] \begin{bmatrix} I_d & \\ & \Pi_d \end{bmatrix} \mathbf{Q}_{2d} \\ &= \mathbf{Q}_m^H [J_1 \mathbf{Q}_M \mathbf{E}_s \ J_2 \mathbf{Q}_M \mathbf{E}_s] \frac{1}{\sqrt{2}} \begin{bmatrix} I_d & jI_d \\ I_d & -jI_d \end{bmatrix} \\ &= \frac{1}{\sqrt{2}} [K_1 \mathbf{E}_s \ K_2 \mathbf{E}_s] \end{aligned}$$

where the selection matrices K_1 and K_2 are defined as follows:

$$\begin{aligned} K_1 &= \mathbf{Q}_m^H (J_1 + \Pi_m J_1 \Pi_M) \mathbf{Q}_M \\ K_2 &= \mathbf{Q}_m^H j (J_1 - \Pi_m J_1 \Pi_M) \mathbf{Q}_M. \end{aligned} \quad (32)$$

Since the matrices in braces are centro-Hermitian, K_1 and K_2 are always real, cf. Theorem 1. They are even sparse if the selection matrix J_1 is sparse and the matrices \mathbf{Q}_m and \mathbf{Q}_M are chosen according to (3) or (4). This is illustrated by the following example. For the ULA with maximum overlap sketched in Fig. 2(a), J_1 is given by

$$J_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{Note: } \begin{array}{l} \text{Pm3, Pm2, Pm1} \\ \text{Qm3, Qm2, Qm1} \\ \text{Qm4, Qm5, Qm6} \end{array}$$

Thus, straightforward calculations yield

$$\begin{aligned} K_1 &= \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \\ K_2 &= \begin{bmatrix} 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & -\sqrt{2} \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

E. Summary of the Algorithm

Before presenting a summary of Unitary ESPRIT, we note an interesting relationship between the eigenvalues of the real matrix \mathbf{T} denoted by ω_k , and the estimated phase factors $\phi_k = e^{j\mu_k}$, cf. (9). Solving $e^{j\mu_k} = f(\omega_k)$ for the spatial frequencies μ_k yields the simple expression

$$\mu_k = \frac{1}{j} \ln \left(\frac{1+j\omega_k}{1-j\omega_k} \right) = 2 \arctan \omega_k, \quad k = 1, 2, \dots, d. \quad (33)$$

Now, we are in the position to summarize the described real implementation of Unitary ESPRIT, which is given in Table I. Here, the left H -real matrices \mathbf{Q}_m and \mathbf{Q}_M are chosen according to (3) or (4).

Notice that a linear estimate of the source signal matrix S (Step 7) can easily be obtained by applying the results of this section to the source signal matrix estimate derived in [4], where (without loss of generality) the additive noise is assumed to be spatially uncorrelated.

V. COMPUTER SIMULATIONS

In this section, we present some simulation results that compare Unitary ESPRIT with the standard ESPRIT algorithm, using the SVD implementation in all cases. Among others, we examine scenarios where the standard ESPRIT algorithm faces some problems, like low signal-to-noise ratios, short window lengths, and correlated source signals.

A. Signal Reconstruction

First, we examine the effect of Unitary ESPRIT on the resulting signal estimates. To this end, three impinging wavefronts are reconstructed using a single ULA of $M = 9$ sensors with maximum overlap, cf. Fig. 2(a). The three uncorrelated equi-powered QPSK signals arrive from $\theta_1 = 10^\circ$, $\theta_2 = 20^\circ$, and $\theta_3 = 30^\circ$, respectively. Fig. 3 depicts the resulting output signal-to-noise-and-interference ratio (SNIR) as a function of the SNR and the number of snapshots N using standard ESPRIT (dashed lines) and Unitary ESPRIT (solid lines). The values of N marked on the right side of the figure correspond to the solid lines, i.e., Unitary ESPRIT. The output SNIR achieved by the standard ESPRIT algorithm for a given value of N (dashed lines) can be found below the corresponding solid lines. For small values of N , e.g., $N = 5$ snapshots, Unitary ESPRIT achieves a significantly better performance than the standard ESPRIT algorithm. Notice that standard ESPRIT with $N = 10$ snapshots attains the same performance as Unitary ESPRIT with $N = 5$ snapshots for SNR's that are greater than 15 dB, while the performance of standard ESPRIT with $N = 20$ is comparable to the performance of Unitary ESPRIT with $N = 10$ for SNR's that are greater than 5 dB. Thus, Unitary ESPRIT essentially doubles the number of available snapshots N compared to the standard ESPRIT algorithm.

B. DOA Estimation

Next, we investigate the effect of Unitary ESPRIT on the estimated phase factors ϕ_k , $1 \leq k \leq d$. Consider a ULA with $M = 6$ sensors and three correlated signals impinging from $\theta_1 = -20^\circ$, $\theta_2 = 0^\circ$, and $\theta_3 = 20^\circ$. Their correlation matrix is given by

$$R_{xx} = \begin{bmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho^2 \\ \rho & \rho^2 & 1 \end{bmatrix}, \quad (34)$$

The phase factors ϕ_1 , ϕ_2 , and ϕ_3 , estimated with the standard ESPRIT algorithm and Unitary ESPRIT, are marked by crosses (\times) in the complex plane as depicted in Figs. 4 and 5 for

TABLE I
SUMMARY OF UNITARY ESPRIT

(1) Initialization:	Form the matrix $\tilde{X} \in \mathbb{C}^{M \times N}$ from the available measurements.
(2) Signal Subspace Estimation:	Determine the real matrix $T(\tilde{X}) \in \mathbb{R}^{M \times 2N}$ from (7) and compute the SVD of $T(\tilde{X})$ (square root approach) or the eigendecomposition of $T(\tilde{X})T(\tilde{X})^H$ (covariance approach). The d dominant left singular vectors or eigenvectors will be called $E_s \in \mathbb{R}^{M \times d}$. Estimate the number of sources d , if d is not known a priori [22].
(3) (Total) Least Squares:	Solve the overdetermined system of equations
	$K_1 E_s T \approx K_2 E_s$
	by means of least squares (or total least squares) techniques. The selection matrices K_1 and K_2 are defined in (32).
(4) Eigenvalue Decomposition:	Compute the eigendecomposition of the resulting solution
	$T = T \Omega T^{-1} \in \mathbb{R}^{d \times d}, \text{ where } \Omega = \text{diag}\{\omega_k\}_{k=1}^d$
(5) Reliability Test:	If all eigenvalues ω_k are real, the estimates will be reliable. Otherwise, start again with more measurements.
(6) DOA Estimation:	Estimate the directions of arrival (DOA's) from
	$\mu_k = 2 \arctan \omega_k \quad k = 1, 2, \dots, d$
	according to (9). $\phi_k = e^{j\mu_k}$
(7) Signal Reconstruction:	A linear estimate of the source signal matrix $S \in \mathbb{C}^{d \times N}$ is given by
	$\hat{S} = (D T^{-1} E_s^H Q_M^H) \tilde{X}_s$
	where $D \in \mathbb{C}^{d \times d}$ denotes an arbitrary diagonal (row) scaling matrix [4].

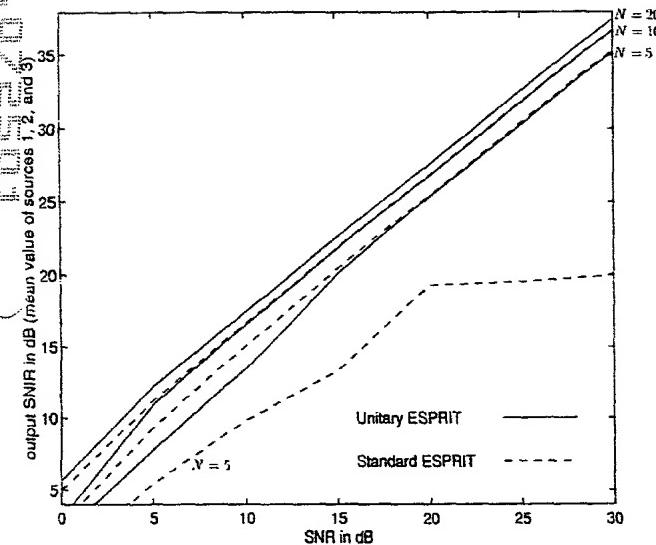


Fig. 3. Output SNIR as a function of the SNR and the number of snapshots N using standard ESPRIT (dashed lines) and Unitary ESPRIT (solid lines) for $\theta_1 = 10^\circ$, $\theta_2 = 20^\circ$, and $\theta_3 = 30^\circ$ ($M = 9$ sensors, 1000 trial runs). The values of N marked on the right side of the figure correspond to the solid lines, i.e., Unitary ESPRIT. The output SNIR achieved by the standard ESPRIT algorithm for a given value of N (dashed lines) can be found below the corresponding solid lines.

a correlation coefficient of $\rho = 0.5$. The results of 80 trial runs with $N = 20$ snapshots and an SNR of 0 dB are shown. Notice that all phase factors estimated with Unitary ESPRIT

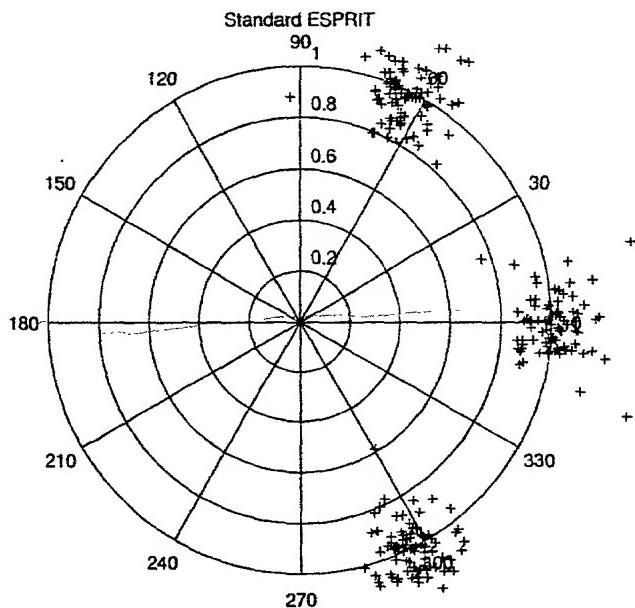


Fig. 4. Phase factors ϕ_1 , ϕ_2 , and ϕ_3 , estimated with the standard ESPRIT algorithm for $\theta_1 = -20^\circ$, $\theta_2 = 0^\circ$, $\theta_3 = 20^\circ$, and correlation coefficients $\rho_{12} = 0.5$, $\rho_{13} = 0.5$, and $\rho_{23} = 0.25$ ($M = 6$ sensors, SNR = 0 dB, $N = 20$, 80 trial runs).

coefficient of $\rho = 0.8$. In this example, the Unitary ESPRIT reliability test has failed three times. To picture these failures, the corresponding phase factor estimates are surrounded by circles (•) of Fig. 7. Notice that the variance of the DOA

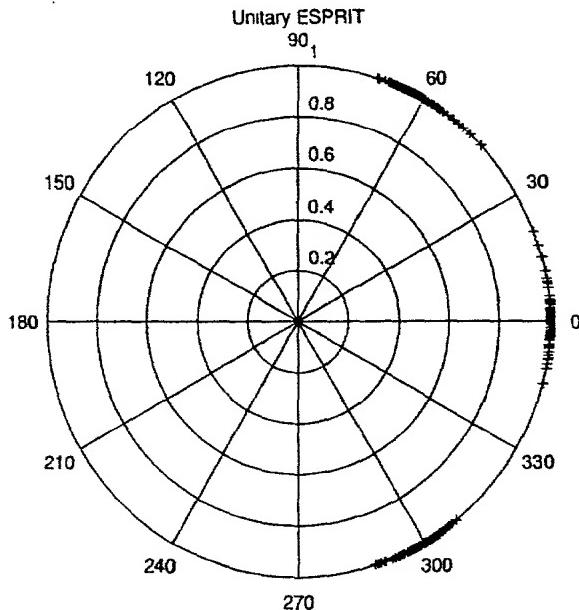


Fig. 5. Phase factors ϕ_1, ϕ_2 , and ϕ_3 , estimated with Unitary ESPRIT for $\theta_1 = -20^\circ, \theta_2 = 0^\circ, \theta_3 = 20^\circ$, and correlation coefficients $\rho_{12} = 0.5, \rho_{13} = 0.5$, and $\rho_{23} = 0.25$ ($M = 6$ sensors, SNR = 0 dB, $N = 20, 80$ trial runs).

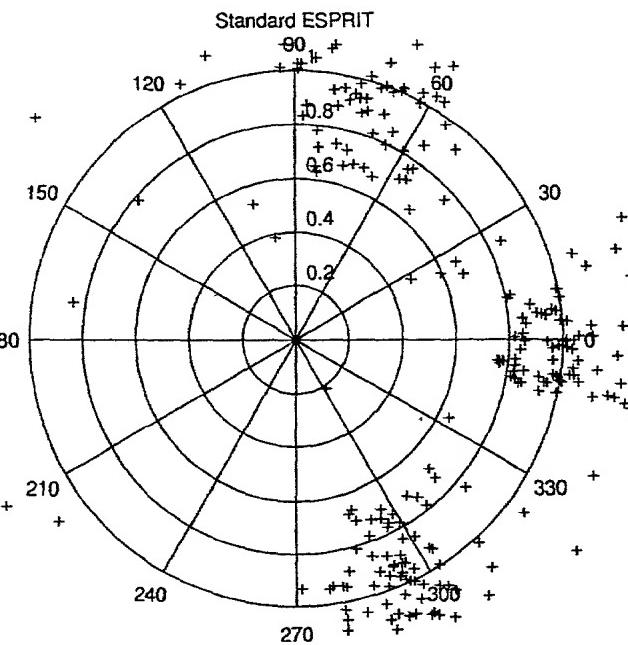


Fig. 6. Phase factors ϕ_1, ϕ_2 , and ϕ_3 , estimated with the standard ESPRIT algorithm for $\theta_1 = -20^\circ, \theta_2 = 0^\circ, \theta_3 = 20^\circ$, and correlation coefficients $\rho_{12} = 0.8, \rho_{13} = 0.8$, and $\rho_{23} = 0.64$ ($M = 6$ sensors, SNR = 0 dB, $N = 20, 80$ trial runs).

lower than the variance of the DOA estimates obtained by the standard ESPRIT approach. The advantages of Unitary ESPRIT become even more evident if the root mean squared error (RMSE) of the estimated directions of arrival is plotted as a function of the correlation coefficient ρ . Fig. 8 show these curves for SNR's of -3, 0, and 5 dB using 3000 trial runs. The standard ESPRIT algorithm (dashed line - - -) is

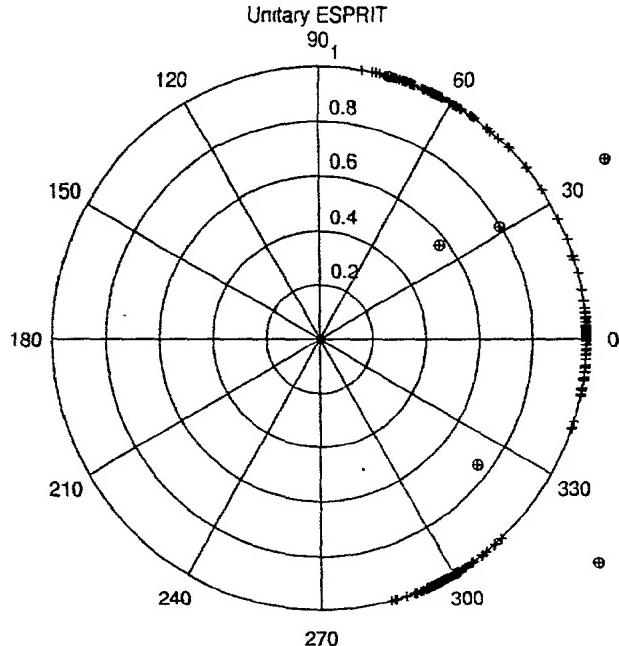


Fig. 7. Phase factors ϕ_1, ϕ_2 , and ϕ_3 , estimated with Unitary ESPRIT for $\theta_1 = -20^\circ, \theta_2 = 0^\circ, \theta_3 = 20^\circ$, and correlation coefficients $\rho_{12} = 0.8, \rho_{13} = 0.8$, and $\rho_{23} = 0.64$ ($M = 6$ sensors, SNR = 0 dB, $N = 20, 80$ trial runs). Estimates that produced a failure of the reliability test are surrounded by a circle (o).

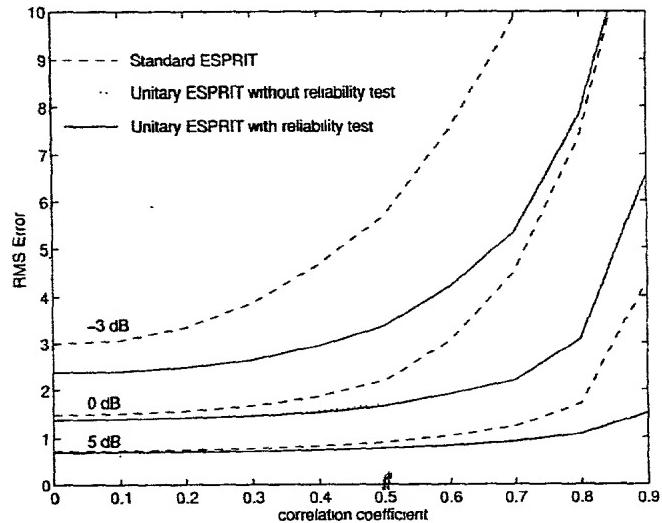


Fig. 8. Root mean squared error (RMSE) in degrees of the estimated directions of arrival as a function of the correlation coefficient ρ and the SNR for $\theta_1 = -20^\circ, \theta_2 = 0^\circ$, and $\theta_3 = 20^\circ$ ($M = 6$ sensors, $N = 20, 3000$ trial runs). The signal correlation matrix is given by (34).

line - - -), and Unitary ESPRIT with the new reliability test (solid line —). It can be seen that Unitary ESPRIT improves the estimation accuracy considerably. In the case of low SNR's, the estimation accuracy is improved even further, by exploiting the information provided by the new reliability test. The corresponding failure rates of the Unitary ESPRIT reliability test are plotted in Fig. 9.

Due to the forward-backward averaging effect, Unitary ESPRIT can separate two completely coherent wavefronts.

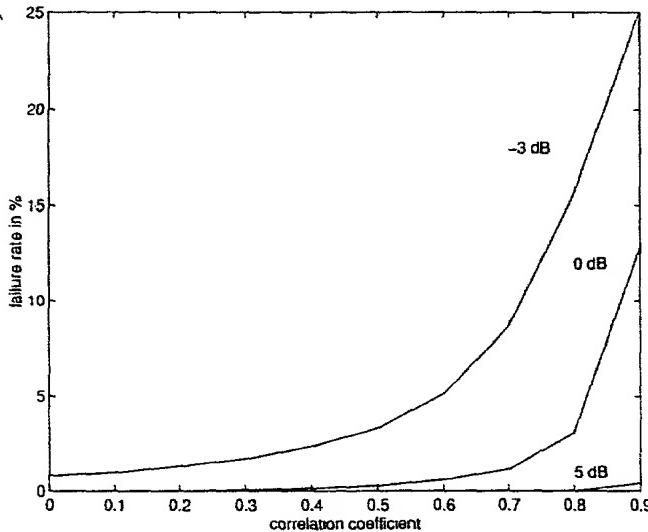


Fig. 9. Failures of the Unitary ESPRIT reliability test as a function of the correlation coefficients ρ for $\theta_1 = -20^\circ$, $\theta_2 = 0^\circ$, and $\theta_3 = 20^\circ$ ($M = 6$ sensors, $N = 20, 3000$ trial runs). Once again, the signal correlation matrix is given by (34). These curves correspond to the solid lines in Fig. 8.

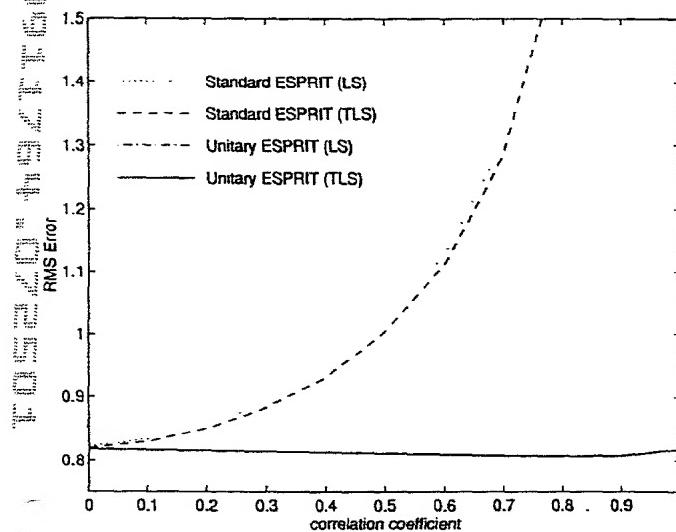


Fig. 10. RMSE (in degrees) of the estimated directions of arrival as a function of the correlation coefficient ρ for $\theta_1 = 0^\circ$ and $\theta_2 = 20^\circ$ ($M = 4$ sensors, $N = 20, 100$ trial runs). Notice that the curves for the LS and the TLS version of Unitary ESPRIT fall on top of one another.

signals with correlation coefficient ρ are impinging an a ULA of $M = 4$ sensors from $\theta_1 = 0^\circ$ and $\theta_2 = 20^\circ$. Fig. 10 shows the resulting RMS error of the estimated DOA's as a function of ρ . The performance of Unitary ESPRIT is not effected by the correlation, while the performance of standard ESPRIT deteriorates dramatically as ρ increases. Notice also that the difference between TLS and LS version of standard ESPRIT is negligible, while the LS and the TLS version of Unitary ESPRIT fall on top of one another. Thus, it is advisable to use the LS version of Unitary ESPRIT instead of the computationally more expensive TLS version. Finally, Figs. 11 and 12 show the RMS error of the estimated DOA's as a function of the magnitude and phase of a complex-valued correlation coefficient ρ , confirming the conclusions drawn

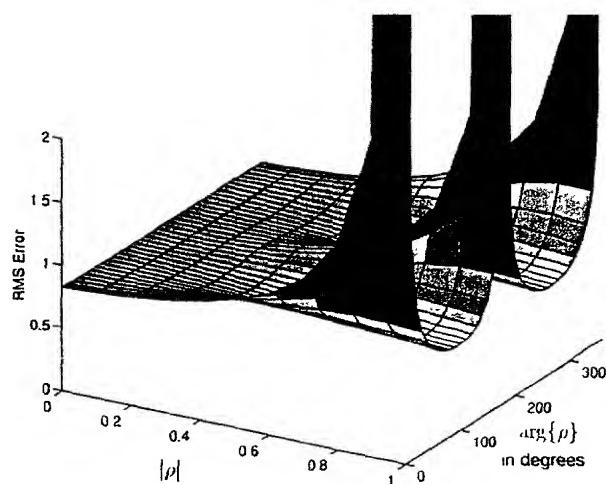


Fig. 11. RMSE (in degrees) of the estimated directions of arrival as a function of the magnitude and phase of the complex correlation coefficient ρ for $\theta_1 = 0^\circ$ and $\theta_2 = 20^\circ$ using standard ESPRIT ($M = 4$ sensors, $N = 20, 100$ trial runs).

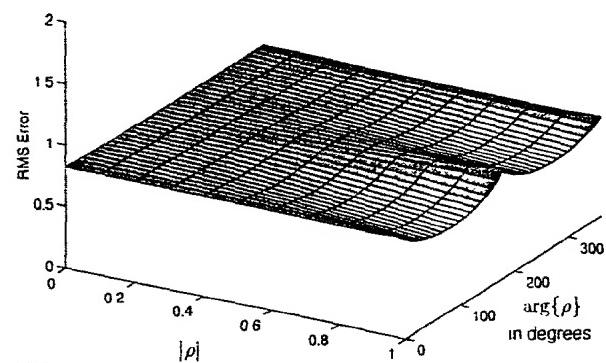


Fig. 12. RMSE (in degrees) of the estimated directions of arrival as a function of the magnitude and phase of the complex correlation coefficient ρ for $\theta_1 = 0^\circ$ and $\theta_2 = 20^\circ$ using Unitary ESPRIT ($M = 4$ sensors, $N = 20, 100$ trial runs).

VI. CONCLUDING REMARKS

An improved version of the ESPRIT algorithm, called Unitary ESPRIT, has been presented in this paper. Unitary ESPRIT represents a simple method to constrain the estimated phase factors to the unit circle, yielding more accurate signal subspace estimates. The computational complexity is reduced significantly by exploiting the one-to-one correspondence between centro-Hermitian and real matrices, allowing a transformation to real matrices, which can be maintained for all steps of the algorithm. Unitary ESPRIT also provides a new reliability test, which is particularly useful in extremely low SNR's. Due to the inherent forward-backward averaging effect, Unitary ESPRIT can separate two completely coherent sources and provides improved estimates for correlated signals. Moreover, Unitary ESPRIT offers a great potential to improve the performance of approximate signal subspace estimation techniques, which are well suited for an adaptive implementation, since inexpensive updating strategies are known [5].

The fact that Unitary ESPRIT is efficiently formulated in

critically important for the extension to 2-D centro-symmetric arrays with a dual invariance structure. 2-D Unitary ESPRIT [23] provides automatically paired source azimuth and elevation angle estimates along with an efficient way to reconstruct the impinging wavefronts. Furthermore, an efficient DFT beampspace implementation of Unitary ESPRIT has also been derived in [23], enabling reduced dimension processing in beampspace, if there is *a priori* information on the general angular location of the DOA's.

ACKNOWLEDGMENT

The authors would like to thank Prof. M. Viberg, Gothenburg, Sweden, for his helpful comments regarding the contents of this paper.

REFERENCES

- [1] C. H. Bischof and G. M. Shroff, "On updating signal subspaces," *IEEE Trans. Signal Processing*, vol. 40, pp. 96–105, Jan. 1992.
- [2] T. F. Chan, "Rank revealing QR factorization," *Linear Algebra and its Applications*, vol. 88, 89, pp. 67–82, 1987.
- [3] J. Götzte and A. J. van der Veen, "On-line subspace estimation using a Schur-type method," *IEEE Trans. Signal Processing*, Nov. 1993, submitted for publication.
- [4] M. Haardt and M. E. Ali-Hackl, "Unitary ESPRIT: How to exploit additional information inherent in the rotational invariance structure," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Adelaide, Australia, Apr. 1994, pp. 229–232, vol. IV.
- [5] M. Haardt and J. Götzte, "Unitary Schur-type subspace estimation," Technical University of Munich, Inst. of Network Theory Circuit Design, Munich, Germany, Tech. Rep. TUM-LNS-TR-94-6, July 1994.
- [6] M. Haardt, P. Weismüller, and R. Killmann, "The identification of late fields: A multichannel high-resolution state space approach," in *Proc. 14th GRETSI Symp. Signal Image Processing*, Juan-les-Pins, France, Sept. 1993, pp. 1251–1254.
- [7] Y. Hua and T. K. Sarkar, "On SVD for estimating generalized eigenvalues of singular matrix pencil in noise," *IEEE Trans. Signal Processing*, vol. 39, pp. 892–900, Apr. 1991.
- [8] K. C. Huang and C. C. Yeh, "A unitary transformation method for angle-of-arrival estimation," *IEEE Trans. Signal Processing*, vol. 39, pp. 975–977, Apr. 1991.
- [9] ———, "Adaptive beamforming with conjugate symmetric weights," *IEEE Trans. Antennas Propagat.*, vol. 39, pp. 926–932, July 1991.
- [10] A. Lee, "Centrohermitian and skew-centrohermitian matrices," *Linear Algebra and its Applications*, vol. 29, pp. 205–210, 1980.
- [11] K. J. R. Liu, D. P. O'Leary, G. W. Stewart, and Y. J. J. Wu, "An adaptive ESPRIT based on URV decomposition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Minneapolis, MN, Apr. 1993, pp. 37–40, vol. IV.
- [12] B. Ottersten, M. Viberg, and T. Kailath, "Performance analysis of the total least squares ESPRIT algorithm," *IEEE Trans. Signal Processing*, vol. 39, pp. 1122–1135, May 1991.
- [13] S. U. Pillai and B. H. Kwon, "Forward/backward spatial smoothing techniques for coherent signal identification," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 8–15, Jan. 1989.
- [14] B. D. Rao and K. V. S. Hari, "Weighted subspace methods and spatial smoothing: Analysis and comparison," *IEEE Trans. Signal Processing*, vol. 41, pp. 788–803, Feb. 1993.
- [15] R. Roy and T. Kailath, "ESPRIT—Estimation of signal parameters via rotational invariance techniques," in *Signal Processing Part II: Control Theory and Applications* (L. Auslander, F. A. Grünbaum, J. W. Helton, T. Kailath, P. Khargonekar, and S. Mitter, Eds.). Berlin, Vienna, New York: Springer-Verlag, 1990, pp. 369–411.
- [16] R. Roy, A. Paulraj, and T. Kailath, "ESPRIT—A subspace rotation approach to estimation of parameters of censored signals in noise," *IEEE Trans.*

- [17] R. H. Roy, "ESPRIT—Estimation of Signal Parameters via Rotational Invariance Techniques," Ph.D. thesis, Stanford Univ., Stanford, CA, Aug. 1987.
- [18] G. W. Stewart, "An updating algorithm for subspace tracking," *IEEE Trans. Signal Processing*, vol. 40, pp. 1535–1541, June 1992.
- [19] A. J. van der Veen, "A Schur method for low-rank matrix approximation," *SIAM J. Matrix Anal. Appl.*, 1994, accepted for publication.
- [20] A. J. van der Veen, E. F. Deprettere, and A. L. Swindlehurst, "Subspace-based signal analysis using singular value decomposition," *Proc. IEEE*, vol. 81, pp. 1277–1308, Sept. 1993.
- [21] S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis Frontiers in Applied Mathematics*, Vol. 9. Philadelphia, PA: Soc. Ind. and Applied Math., 1991.
- [22] G. Xu, R. H. Roy, and T. Kailath, "Detection of number of sources via exploitation of centro-symmetry property," *IEEE Trans. Signal Processing*, vol. 42, pp. 102–112, Jan. 1994.
- [23] M. D. Zoltowski, M. Haardt, and C. P. Mathews, "Closed-form angle estimation with rectangular arrays in element space or beampspace via Unitary ESPRIT," *IEEE Trans. Signal Processing*, July 1994, submitted for publication.
- [24] M. D. Zoltowski, G. M. Kautz, and S. D. Silverstein, "Beampspace Root-MUSIC," *IEEE Trans. Signal Processing*, vol. 41, pp. 344–364, Jan. 1993.



Martin Haardt (S'90) was born in Germany on October 4, 1967 and received the Diplom-Ingenieur degree in electrical engineering from the Ruhr-University Bochum, Germany, in 1991.

From 1989 to 1990, he was a visiting scholar at Purdue University, West Lafayette, IN, USA, sponsored by a fellowship of the German Academic Exchange Service (DAAD). From 1991 to 1993, he worked for Siemens AG, Corporate Research and Development, in Munich, Germany, conducting research in the areas of image processing and biomedical signal processing. Since 1993 he has been a research associate with the Institute of Network Theory and Circuit Design at the Technical University of Munich, Germany, where he is also working toward a Ph.D. degree. His current research interests include array signal processing, spectral estimation, numerical linear algebra, and mobile communications.

Josef A. Nossek (S'72–M'74–SM'81–F'93) was born in Vienna, Austria, on December 17, 1947 and received the Diplom-Ingenieur and the Ph.D. degrees, both in electrical engineering, from the Technical University of Vienna, Austria, in 1974 and 1980, respectively.

In 1974, he joined Siemens AG, Munich, Germany, where he was engaged in the design of passive and active filters for communication systems. In 1978, he became a supervisor, and in 1980 the head of a group of laboratories concerned with the design of monolithic filters (analog and digital) and electromechanical and microwave filters. In 1982, he became head of a group of laboratories designing digital radio systems within the Transmission Systems Department. In 1985, he spent a month as a visiting professor at the University of Cape Town, South Africa. From 1987 to 1989, he was head of the Radio Systems Design Department. Since April 1989, he has been a full professor and head of the Institute of Network Theory and Circuit Design at the Technical University of Munich, Germany. He is teaching undergraduate and graduate courses in the field of circuit and system theory and conducting research in the areas of real-time signal processing, neural networks, and dedicated VLSI-architectures.

Dr. Nossek has published more than 100 papers in scientific and technical journals and conference proceedings. He holds a number of patents. In 1988, he received the ITG best paper award. From 1991 to 1993, he served as an associate editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS.

$$\text{of (5). } Q_p^{*\top} M Q_q = V_p \Sigma_p V_p^{*\top} \Rightarrow (Q_p^{*\top})^{-1} V_p =$$

$$\Sigma_p V_p^{*\top} (Q_q)^{-1} = M \Rightarrow$$

Q_p, Q_q unitary

$$(Q_p V_p) \Sigma_p (V_p^{*\top} Q_q) = M \quad (5)$$

M : centrohermitian, i.e. $\Pi_p M^* \Pi_q = M$

Ref. [5]

Joint Detection with Coherent Receiver Antenna Diversity in CDMA Mobile Radio Systems

Peter Jung, Member, IEEE, and Josef Blanz

J1017 U.S. PRO
09/911764
07/25/01

Abstract— In code division multiple access (CDMA) mobile radio systems, both intersymbol interference and multiple access interference arise which can be combated by using either elaborate optimum or favorable suboptimum joint detection (JD) techniques. Furthermore, the time variation of the radio channels leads to degradations of the receiver performance. These degradations can be reduced by applying diversity techniques. Using coherent receiver antenna diversity (CRAD) is especially attractive because only the signal processing at the receiver must be modified. In the present paper, the application of CRAD to the more critical uplink of CDMA mobile radio systems with suboptimum JD techniques is investigated for maximal-ratio combining. The authors study six different suboptimum JD techniques based on decorrelating matched filtering, Gauss-Markov estimation, and minimum mean square error estimation with and without decision feedback. These six suboptimum JD techniques which are well-known for single antenna receivers are extended for the application to CRAD. A main concern of the paper is the determining of the SNR performance of the presented JD techniques for CRAD and the achievable average uncoded bit error probabilities for the transmission over rural area, typical suburban and bad urban mobile radio channels are determined.

I. INTRODUCTION

TWO MAJOR PROBLEMS must be solved in digital mobile radio systems, namely the multiple access (MA) problem, arising from the simultaneous transmission of signals associated with several active users that share the same transmission medium, and the equalization problem, arising from time-varying and frequency-selective mobile radio channels [1], [2]. An attractive solution to the MA problem is code division multiple access (CDMA) [1] which is currently under discussion for the application to third generation digital mobile radio systems [3]–[7]. In a CDMA mobile radio system, multiple access interference (MAI) between data symbols of different users occurs which can be combated successfully by applying either joint detection (JD) [4], [8], [9] or interference cancellation (IC) [10] techniques. Furthermore, intersymbol interference (ISI) arises between consecutive data symbols associated with a single user when CDMA is used due to the frequency selectivity of mobile radio channels. ISI is negligible when the symbol period T_s is much greater than the duration of the channel impulse response. However, ISI becomes severe when T_s is lower than or approximately equal to the duration of the channel impulse response. With respect to generality, the existence of ISI in conjunction with MAI in CDMA mobile

radio systems is assumed in the following. Therefore, a major requirement for detection in such CDMA mobile radio systems is the combatting of both MAI as well as ISI, respectively.

In this paper, JD techniques fulfilling the aforementioned requirement such as presented in [4], [8], [9] shall be considered. Unfortunately, the maximum likelihood JD technique of [8] is prohibitively expensive. Therefore, suboptimum JD techniques [3], [4], [9], [11]–[14] are more favorable. These JD techniques are appropriately modified versions of well known adaptive equalizers such as reviewed by Qureshi in his famous paper [15].

Many mobile radio systems such as the successful pan-European Global System for Mobile Communications (GSM) [16], [17], the Digital Cellular System (DCS) 1800 [18], the Japanese Digital Cellular (JDC) [17] and the American Digital Cellular (ADC), also termed IS-54, apply discontinuous, i.e., burst, transmission. These systems, therefore use time division multiple access (TDMA) combined with frequency division multiple access (FDMA). Burst transmission is advantageous with respect to the implementation of the receiver hardware, cf., i.e. [16], [17] for a detailed rationale. Therefore, it stands to reason to develop novel mobile radio systems in an evolutionary manner by introducing an additional CDMA component to the aforementioned F/TDMA based mobile radio systems in order to achieve a capacity improvement over the aforementioned F/TDMA based mobile radio systems. Such a CDMA mobile radio system using FDMA and TDMA and applying JD techniques which was evolved from GSM was proposed in [19]. This system shall be termed JD-CDMA mobile radio system in what follows. Following the report of [19], burst transmission shall be considered in what follows. With respect to a moderate hardware expense of the receivers for the JD-CDMA mobile radio system, synchronization between the users, i.e., mobiles and base stations, is favorable and can be easily achieved based on the TDMA component, i.e. by following the procedures used in GSM, cf. [20]. However, the problem of synchronization shall not be covered in this paper. Furthermore, an isolated cell of the cellular JD-CDMA mobile radio system environment is considered in what follows. Intercell interference is treated as additional noise.

In the synchronous JD-CDMA mobile radio system, the signal processing for the downlink and for the uplink differ considerably. In the case of the downlink, the signals associated with K simultaneously active users are radiated from the same location, namely the base station. Hence, all K user

which are radiated from K separate mobiles, is associated with K separate radio channels. Therefore, the separation of the K user signals is more critical in the uplink than in the downlink. The authors consider only the uplink in the following.

Conventionally, matched filtering, in particular by applying decorrelating matched filters (DMF's), or correlation including RAKE concepts are applied in CDMA mobile radio systems in order to accomplish the suboptimum data detection [1], [7], [21]. Matched filtering is often inefficient because a tight power control is required and the MAI is treated as noise. However, the data detection with DMF's can be regarded as a JD technique which shall be done in what follows. Setting out from the DMF, decision feedback equalizers which are similar to interference cancellation techniques, cf. [10], [22], can be implemented. Such equalizers shall be termed decorrelating matched filter block decision feedback equalizers (DMF-BDFE) in what follows. Further well known suboptimum JD techniques for single antenna receivers applicable to a synchronous JD-CDMA mobile radio system are the zero forcing block linear equalizer (ZF-BLE) [3], [4], [11], [12] based on the Gauss-Markov estimation [23] and the minimum mean square error [23], [24] block linear equalizer (MMSE-BLE) [3], [12]. Setting out from the ZF-BLE and the MMSE-BLE, block decision feedback equalizers referred to as zero forcing block decision feedback equalizer (ZF-BDFE) and minimum mean square error block decision feedback equalizer (MMSE-BDFE), respectively, were developed for single antenna receivers [3], [11], [12].

The time variation of the radio channels leads to a considerable variation of the signal-to-noise ratio (SNR) at the detector input. With respect to a constant expectation of the SNR, the error probability increases with increasing variation of the SNR [2]. Hence, the reduction of this variation of the SNR, i.e. by using diversity techniques [25], is a desirable asset. The application of receiver antenna diversity [25] is especially favorable because only the signal processing at the receiver must be modified whereas the signaling scheme of a synchronous JD-CDMA mobile radio system remains unaffected. In the course of the paper, the authors introduce novel representations of the JD techniques DMF, DMF-BDFE, ZF-BLE, ZF-BDFE, MMSE-BLE, and MMSE-BDFE for coherent receiver antenna diversity (CRAD) and maximal-ratio combining [25]. Furthermore, it shall be shown that all six JD techniques are special cases of a basic concept for JD. A main concern of the paper is the determining of the SNR performance of these six JD techniques for CRAD and the achievable uncoded bit error probabilities for the transmission over rural area, typical urban, and bad urban mobile radio channels by using the channel models defined by COST 207 [26], [27] are determined.

The paper is organized as follows. In Section II, the uplink model of the JD-CDMA mobile radio system with CRAD which is to be investigated is introduced. The extension of the DMF, the DMF-BDFE, the ZF-BLE, the ZF-BDFE, the MMSE-BLE, and the MMSE-BDFE from the single antenna case to CRAD as well as the corresponding SNR performances are presented in Section III. In Section IV, the

transmission over rural area, typical urban, and bad urban mobile radio channels are given.

In the present paper, vectors and matrices are in bold face. Furthermore, the symbols $(\cdot)^*$, $(\cdot)^T$, $\|\cdot\|$, and $E\{\cdot\}$ designate the complex conjugation, the transposition, the vector norm, and the expectation, respectively.

II. SYSTEM MODEL

The discrete-time lowpass representation of the uplink in a synchronous JD-CDMA mobile radio system with CRAD shall be described in this section. The basic structure of the uplink is depicted in Fig. 1. It is assumed that K users, i.e. the mobiles, are simultaneously active in the same frequency band transmitting the finite data symbol sequences

$$\mathbf{d}^{(k)} = (d_1^{(k)}, d_2^{(k)}, \dots, d_N^{(k)})^T, \quad d_n^{(k)} \in V, \quad V \subset \mathbb{C}, \\ k = 1, \dots, K, \quad n = 1, \dots, N, \quad K, N \in \mathbb{N} \quad (1)$$

of N m -ary complex data symbols $d_n^{(k)}$ with period T_s which are taken from the complex set

$$V = \{v_1, v_2, \dots, v_m\}, \\ v_\mu \in \mathbb{C}, \mu = 1, \dots, m, \quad m \in \mathbb{N}. \quad (2)$$

The actual data rate may be varied due to the service provided or due to the desired transmission quality. Each of the data symbols $d_n^{(k)}$, $n = 1, \dots, N$, of mobile k is spread by using the user specific signature sequence

$$\mathbf{c}^{(k)} = (c_1^{(k)}, c_2^{(k)}, \dots, c_Q^{(k)})^T, \quad c_q^{(k)} \in V_c, \quad V_c \subset \mathbb{C}, \\ k = 1, \dots, K, \quad q = 1, \dots, Q, \quad K, Q \in \mathbb{N} \quad (3)$$

at the transmitters in order to allow a coexistence of K simultaneously transmitted signals in the same frequency band. The m -ary complex signature elements $c_q^{(k)}$ of (3) which are taken from the complex set

$$V_c = \{v_{c,1}, v_{c,2}, \dots, v_{c,m}\}, \quad v_{c,\mu} \in \mathbb{C}, \\ \mu = 1, \dots, \bar{m}, \quad \bar{m} \in \mathbb{N} \quad (4)$$

are termed chips. The chip duration is given by

$$T_c = \frac{T_s}{Q}. \quad (5)$$

Each mobile is assumed to have a single transmitter-antenna. The transmitted signals are received at the uplink receiver, i.e. the base station, over K_a receiver antennas. Therefore, the transmission of the K user signals takes place over $K \cdot K_a$ different radio channels with the time-variant complex impulse responses

$$h^{(k,k_a)}(\tau, t), \quad k = 1, \dots, K, \quad k_a = 1, \dots, K_a, \quad K, K_a \in \mathbb{N} \quad (6)$$

where the radio channel with the impulse response $h^{(k,k_a)}(\tau, t)$ refers to the connection of mobile k with receiver antenna k_a . In (6), τ denotes the delay parameter referring to the time spreading, i.e. distortion, of the transmitted signals due to multipath reception, and t denotes the real time referring to

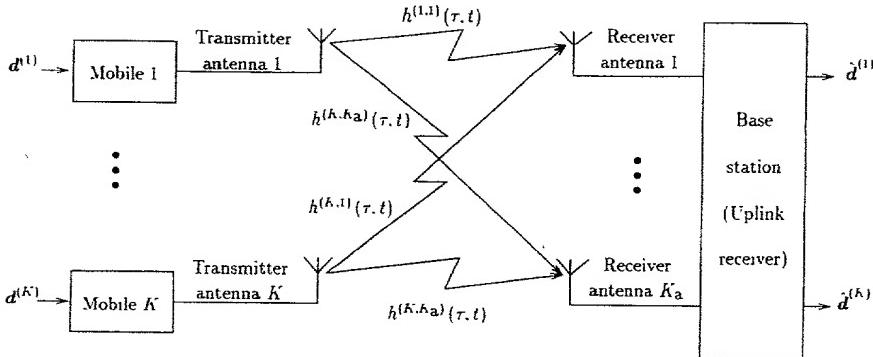


Fig. 1. Structure of the uplink of a JD-CDMA mobile radio system with CRAD.

$K \cdot K_a$ radio channels are assumed to be wide sense stationary uncorrelated scattering (WSSUS) multipath channels [26], and therefore the simulation models of these radio channels can be based on the COST 207 specifications [26], [27].

In the simulation programs, the $K \cdot K_a$ WSSUS multipath channels are implemented in accordance with the Monte Carlo-based model introduced by Schulze [28] and further investigated by Höher [26]. This simple statistical model is entirely determined by the two-dimensional scattering function $S(\tau, f_D)$ which is only dependent on the delay parameter τ and the Doppler frequency f_D associated with the motion of the mobiles. In the following, a brief overview of this statistical model shall be given. In the statistical model used in the simulations, the distribution of the Doppler frequencies f_D is assumed to be independent of the distribution of the delay parameter τ . Therefore, the scattering function can be displayed as the product of the Doppler power spectrum $S_c(0, f_D)$ which only depends on the Doppler frequency f_D and the delay power spectrum $\varrho_T(\tau, 0)$ which only depends on the delay parameter τ , i.e.,

$$S(\tau, f_D) = S_c(0, f_D) \cdot \varrho_T(\tau, 0) \quad (7)$$

holds. The $K \cdot K_a$ channel impulse responses $h^{(k,k_a)}(\tau, t)$ of (6) are samples of $K \cdot K_a$ bandlimited stochastic Gaussian processes which are assumed to be ergodic. Furthermore, only Rayleigh fading is considered throughout the paper. In the digital simulations, each $h^{(k,k_a)}(\tau, t)$ is the linear superposition of P , $P \in \mathbb{N}$, uncorrelated echoes, each with a particular delay parameter $\tau_p^{(k,k_a)}$, $p = 1, \dots, P$, which are represented by P sinusoids with equally distributed phase angles $\theta_p^{(k,k_a)} \in [0, 2\pi]$, $p = 1, \dots, P$, and Doppler frequencies $f_{D,p}^{(k,k_a)}$, $p = 1, \dots, P$, being distributed according to the desired Doppler power spectra associated with the aforementioned bandlimited stochastic Gaussian processes. The Doppler power spectra are assumed to be the classical Doppler power spectra [29, ch. 1.2.1] in this work. The delay parameters $\tau_p^{(k,k_a)}$ are exponentially distributed according to the delay power spectra associated with the desired environment types rural area (RA), bad urban (BU) or typical urban (TU) [27]. With $\delta(\cdot)$ designating Dirac's delta-function, the $K \cdot K_a$

channel impulse responses are therefore given by

$$h^{(k,k_a)}(\tau, t) = \lim_{P \rightarrow \infty} \frac{1}{\sqrt{P}} \sum_{p=1}^P \exp\{j\theta_p^{(k,k_a)}\} \cdot \exp\{j2\pi f_{D,p}^{(k,k_a)} t\} \cdot \delta(\tau - \tau_p^{(k,k_a)}) \quad (8)$$

[26]. A phase modulation fading simulator can be realized by implementing (8), cf. [29, Fig. 1.7-1(a)]. The factor $1/\sqrt{P}$ ensures the convergence of (8) in the limit $P \rightarrow \infty$ [26]. The Gaussian distribution of the quadrature components of $h^{(k,k_a)}(\tau, t)$ defined by (8) follows from the central limit theorem provided that P is sufficiently large. In the simulations, P equal to 600 is chosen. In (8), the amplitudes of the various sinusoids are all equal to $1/\sqrt{P}$ which is reasonable because the simulation model of the radio channels is statistically fully determined by the two-dimensional scattering function [26].

In Fig. 2, a typical simulated plot of the absolute value $|h^{(k,k_a)}(\tau, t)|$ for arbitrary k and k_a is depicted versus τ and t in the case of BU and a vehicle speed equal to 50 km/h. The delay parameter τ is resolved with $0.30 \mu s$ in Fig. 2. In order to relate the duration of $h^{(k,k_a)}(\tau, t)$ to the symbol period T_s , τ is given in units of T_s with T_s equal to $7 \mu s$ which is used in the simulations discussed in Section IV. A new sample of $|h^{(k,k_a)}(\tau, t)|$ is sketched every $100 \mu s$ along the t -axis. The time t is depicted in the range 0 to 8 ms in all cases. The time variation of $h^{(k,k_a)}(\tau, t)$ is obvious from Fig. 2. The duration of $h^{(k,k_a)}(\tau, t)$ shown in Fig. 2 is about $10 \mu s$, and almost 30 distinct paths occur. It is thus evident that the occurrence of ISI is inevitable at the detector in the case of BU. Furthermore, the strengths of the different and independently fading paths are in accordance with the scattering function defined in [27] for BU radio channels.

At the base station, the estimates

$$\hat{\mathbf{d}}^{(k)} = (\hat{d}_1^{(k)}, \hat{d}_2^{(k)}, \dots, \hat{d}_N^{(k)})^T, \quad \hat{d}_n^{(k)} \in V, \quad V \subset \mathbb{C}, \\ k = 1, \dots, K, \quad n = 1, \dots, N, \quad K, N \in \mathbb{N} \quad (9)$$

of the data symbol sequences $\mathbf{d}^{(k)}$ defined by (1) must be determined.

Setting out from Fig. 1, a discrete-time model of the uplink in a synchronous JD-CDMA mobile radio system with CRAD can be derived in analogy to the single antenna receiver case [3], [4], [11], [12]. Each of the $K \cdot K_a$ resulting discrete-time

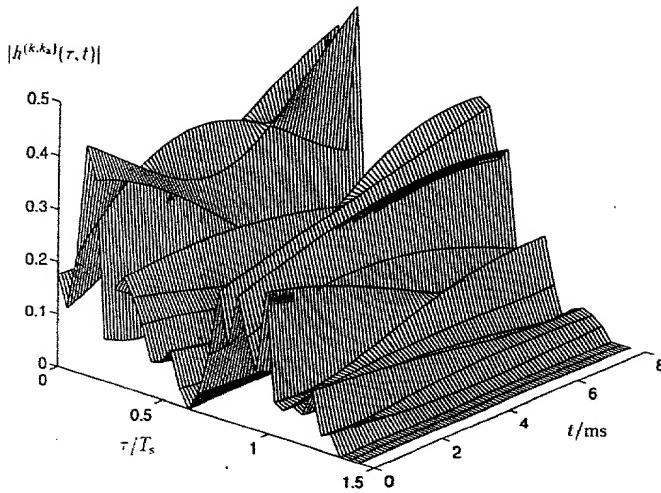


Fig. 2. Absolute value $|H^{(k,k_a)}(\tau, t)|$ of the simulated complex impulse response versus τ and t for arbitrarily chosen k and k_a , in the case of BU and a vehicle speed of 50 km/h.

radio channels is represented by its channel impulse response

$$\mathbf{h}^{(k,k_a)} = (h_1^{(k,k_a)}, h_2^{(k,k_a)}, \dots, h_W^{(k,k_a)})^T, \\ h_w^{(k,k_a)} \in \mathbb{C}, k = 1, \dots, K, k_a = 1, \dots, K_a, \\ w = 1, \dots, W, K, K_a, W \in \mathbb{N} \quad (10)$$

consisting of W complex samples $h_w^{(k,k_a)}$ taken at the chip rate $1/T_c$. In the discrete-time model, the data symbol sequences $\mathbf{d}^{(k)}$ given in (1) are transmitted over $K \cdot K_a$ discrete-time channels with the combined channel impulse responses

$$\mathbf{b}^{(k,k_a)} = (b_1^{(k,k_a)}, b_2^{(k,k_a)}, \dots, b_{Q+W-1}^{(k,k_a)})^T = \mathbf{h}^{(k,k_a)} * \mathbf{c}^{(k)}, \\ b_l^{(k,k_a)} \in \mathbb{C}, k = 1, \dots, K, k_a = 1, \dots, K_a, \\ l = 1, \dots, Q + W - 1, K, K_a, Q, W \in \mathbb{N} \quad (11)$$

consisting of the discrete-time convolution of $\mathbf{h}^{(k,k_a)}$ introduced in (10) with the signature sequences $\mathbf{c}^{(k)}$ defined by (3). ISI arise for W greater than one and MAI occur for W greater than one and/or for nonorthogonal signature sequences $\mathbf{c}^{(k)}$. The discrete-time channels with the combined channel impulse responses $\mathbf{b}^{(k,k_a)}$, $k = 1, \dots, K$, $k_a = 1, \dots, K_a$, according to (11) are called $(Q + W - 1)$ -path channels. In what follows, it is assumed that the combined channel impulse responses $\mathbf{b}^{(k,k_a)}$ introduced in (11) are known at the receiver which can be guaranteed by using perfect channel estimation. Channel estimation shall not be considered in what follows because the effect of channel estimation errors on the data detection is not of interest when investigating the potential of different data detectors. The reader is referred to [20], instead, where channel estimation algorithms applicable to the considered JD-CDMA mobile radio system with CRAD as well as their effect on the data detection are studied in detail.

Fig. 3 shows a schematic of the uplink in a JD-CDMA mobile radio system with CRAD. The $(Q + W - 1)$ -path channels are modeled as finite impulse response (FIR) filters. Each FIR filter consists of a shift register with $(Q + W - 2)$

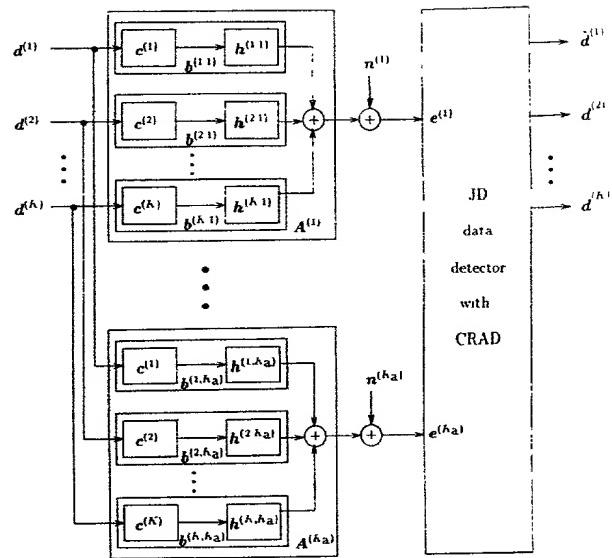


Fig. 3. Discrete-time model of the uplink of a JD-CDMA mobile radio system with CRAD.

Firstly, the well-known case of a single receiver antenna with label k_a is considered [4]. According to Fig. 3, the received sequence $\mathbf{e}^{(k_a)}$ of length $(N \cdot Q + W - 1)$ prevails at antenna k_a . Equivalently to the single antenna receiver discussed in [4], each received sequence $\mathbf{e}^{(k_a)}$ consists of a sum of K sequences, each of length $(N \cdot Q + W - 1)$ and containing the data symbol sequences $\mathbf{d}^{(k)}$ introduced in (1), which are perturbed by an additive stationary noise sequence

$$\mathbf{n}^{(k_a)} = (n_1^{(k_a)}, n_2^{(k_a)}, \dots, n_{N \cdot Q + W - 1}^{(k_a)})^T, \\ n_n^{(k_a)} \in \mathbb{C}, k_a = 1, \dots, K_a, \\ n = 1, \dots, N \cdot Q + W - 1, k_a, N, Q, W \in \mathbb{N} \quad (12)$$

with zero mean and covariance matrix [24], [30]

$$\mathbf{R}_n^{(k_a)(k_a)} = E\{\mathbf{n}^{(k_a)} \mathbf{n}^{(k_a)*T}\}, k_a = 1, \dots, K_a, K_a \in \mathbb{N}. \quad (13)$$

Introducing the data vector

$$\mathbf{d} = (\mathbf{d}^{(1)T}, \mathbf{d}^{(2)T}, \dots, \mathbf{d}^{(K)T})^T = (d_1, d_2, \dots, d_{K \cdot N})^T, \\ K, N \in \mathbb{N} \quad (14)$$

where the components of \mathbf{d} are given by

$$d_{N \cdot (k-1)+n} \stackrel{\text{def}}{=} d_n^{(k)}, k = 1, \dots, K, n = 1, \dots, N, \\ K, N \in \mathbb{N} \quad (15)$$

and defining the $(N \cdot Q + W - 1) \times K \cdot N$ -matrix

$$\mathbf{A}^{(k_a)} = (A_{i,j}^{(k_a)}), i = 1, \dots, N \cdot Q + W - 1, \\ j = 1, \dots, K \cdot N, k_a = 1, \dots, K_a, \quad (16a)$$

$$A_{Q(n-1)+l, N \cdot (k-1)+n}^{(k_a)} = \begin{cases} b_l^{(k,k_a)} & \text{for } k = 1, \dots, K, \\ & k_a = 1, \dots, K_a, \\ & l = 1, \dots, Q + W - 1, \\ & n = 1, \dots, N. \end{cases}$$

the received sequence can be represented by

$$\mathbf{e}^{(k_a)} = (e_1^{(k_a)}, e_2^{(k_a)}, \dots, e_{N \cdot Q + W - 1}^{(k_a)})^T = \mathbf{A}^{(k_a)} \mathbf{d} + \mathbf{n}^{(k_a)}. \quad k_a = 1, \dots, K_a, K_a, N, Q, W \in \mathbb{N}. \quad (17)$$

Now, the unified mathematical representation for K_a receiver antennas is presented. With the $K_a \times (N \cdot Q + W - 1) \times K \cdot N$ -matrix

$$\mathbf{A} = (A_{i,j}) = (\mathbf{A}^{(1)T}, \mathbf{A}^{(2)T}, \dots, \mathbf{A}^{(K_a)T})^T, \quad K_a \in \mathbb{N} \quad (18)$$

and with the combined noise vector

$$\begin{aligned} \mathbf{n} &= (\mathbf{n}^{(1)T}, \mathbf{n}^{(2)T}, \dots, \mathbf{n}^{(K_a)T})^T \\ &= (n_1, n_2, \dots, n_{K_a \cdot (N \cdot Q + W - 1)})^T. \end{aligned} \quad K_a, N, Q, W \in \mathbb{N} \quad (19)$$

where

$$\begin{aligned} n_{(N \cdot Q + W - 1) \cdot (k_a - 1) + n} &\stackrel{\text{def}}{=} n_n^{(k_a)}, \quad k_a = 1, \dots, K_a. \\ n &= 1, \dots, N \cdot Q + W - 1, \quad k_a, N, Q, W \in \mathbb{N} \end{aligned} \quad (20)$$

holds, having the covariance matrix

$$\begin{aligned} \mathbf{R}_n &= E\{\mathbf{n}\mathbf{n}^{*T}\} \\ &= \begin{pmatrix} \mathbf{R}_n^{(1)(1)} & \mathbf{R}_n^{(1)(2)}, \dots, \mathbf{R}_n^{(1)(K_a)} \\ \mathbf{R}_n^{(2)(1)} & \mathbf{R}_n^{(2)(2)}, \dots, \mathbf{R}_n^{(2)(K_a)} \\ \vdots & \vdots \quad \ddots \quad \vdots \\ \mathbf{R}_n^{(K_a)(1)} & \mathbf{R}_n^{(K_a)(2)}, \dots, \mathbf{R}_n^{(K_a)(K_a)} \end{pmatrix}. \end{aligned} \quad (21a)$$

$$\mathbf{R}_n^{(i)(j)} = E\{\mathbf{n}^{(i)}\mathbf{n}^{(j)*T}\}, \quad i, j = 1, \dots, K_a, K_a \in \mathbb{N} \quad (21b)$$

the combined received vector is given by

$$\begin{aligned} \mathbf{e} &= (\mathbf{e}^{(1)T}, \mathbf{e}^{(2)T}, \dots, \mathbf{e}^{(K_a)T})^T = (e_1, e_2, \dots, e_{K_a \cdot (N \cdot Q + W - 1)})^T \\ &= \mathbf{Ad} + \mathbf{n}, \quad K_a, N, Q, W \in \mathbb{N} \end{aligned} \quad (22)$$

with

$$\begin{aligned} e_{(N \cdot Q + W - 1) \cdot (k_a - 1) + n} &\stackrel{\text{def}}{=} e_n^{(k_a)}, \quad k_a = 1, \dots, K_a. \\ n &= 1, \dots, N \cdot Q + W - 1, \quad k_a, N, Q, W \in \mathbb{N}. \end{aligned} \quad (23)$$

The combined received vector \mathbf{e} of (22) is processed in a data detector in order to determine the estimates $\hat{\mathbf{d}}^{(k)}$ defined by (9).

III. JOINT DETECTION TECHNIQUES FOR COHERENT RECEIVER ANTENNA DIVERSITY

A. Basic Concept

In what follows, it is assumed that the received vector \mathbf{e} defined by (22) is completely known at the receiver before the data detection is carried out. The basic concept of the JD techniques considered in this paper is given by the set of equations

$$\mathbf{S}\hat{\mathbf{d}} = \mathbf{M}\mathbf{e} \quad (24)$$

where

$$\mathbf{S} = (S_{i,j}), \quad i, j = 1, \dots, K \cdot N \quad (25)$$

is a square matrix with $K \cdot N$ rows and $K \cdot N$ columns,

$$\begin{aligned} \mathbf{M} &= (M_{i,j}), \quad i = 1, \dots, K \cdot N, \\ j &= 1, \dots, K_a \cdot (N \cdot Q + W - 1) \end{aligned} \quad (26)$$

is a $K \cdot N \times K_a \cdot (N \cdot Q + W - 1)$ -matrix and

$$\hat{\mathbf{d}} = (\hat{d}_1, \hat{d}_2, \dots, \hat{d}_{K \cdot N})^T \quad (27)$$

is the estimate of \mathbf{d} defined by (14). The choice of the matrices \mathbf{M} and \mathbf{S} determines the equalizer type. The estimated data symbols \hat{d}_n contained in $\hat{\mathbf{d}}$ of (27) are either continuous valued, referred to as $\hat{d}_{c,n}$ contained in $\hat{\mathbf{d}}_c$, or discrete-valued, referred to as $\hat{d}_{q,n}$ contained in $\hat{\mathbf{d}}_q$. The continuous valued estimates $\hat{d}_{c,n}$ must be quantized [4] in order to yield the desired discrete-valued estimates $\hat{d}_n^{(k)}$ introduced in Section II. Furthermore, it is assumed that $(\hat{\mathbf{d}}^{(1)T}, \hat{\mathbf{d}}^{(2)T}, \dots, \hat{\mathbf{d}}^{(K)T})^T$ introduced in Section II is always identical with the discrete-valued estimate \mathbf{d}_q .

In the following, the representations of \mathbf{M} and \mathbf{S} in the case of CRAD are introduced for the DMF, the DMF-BDFE, the ZF-BLE, the ZF-BDFE, the MMSE-BLE, and the MMSE-BDFE. It shall be shown for the latter five JD techniques that \mathbf{M} always contain a DMF. The transmission of data symbols $d_n^{(k)}$ with $E\{d_n^{(k)}\}$ equal to zero is assumed in what follows. In this case the SNR

$$\begin{aligned} \gamma(k, n) &= \frac{\text{variance of the signal component}}{\text{variance of the noise and interference components}} \\ &= \frac{E\left\{\left|\hat{d}_{N \cdot (k-1)+n}\right|^2\right\}|_{\text{noiseless, } d_n^{(k)} \text{ isolated}}}{E\left\{\left|\hat{d}_{N \cdot (k-1)+n}\right|^2\right\}|_{d_n^{(k)}=0}}. \\ k &= 1, \dots, K, \quad n = 1, \dots, \mathbb{N} \end{aligned} \quad (28)$$

associated with the estimate $\hat{d}_{N \cdot (k-1)+n}$ of the data symbol $d_n^{(k)}$ transmitted by user k at time nT_s at the output of the considered equalizers is given.

B. Decorrelating Matched Filter

Conventionally, the data detection in a JD-CDMA mobile radio system for a single antenna receiver is accomplished by using a DMF followed by symbol-rate samplers, cf. Section I. The extension of the DMF to CRAD is straightforward by using the definitions for \mathbf{A} , \mathbf{R}_n and \mathbf{e} introduced in Section II. With the notation $\text{Diag}(X_{i,i})$ [32] denoting a diagonal matrix containing only the diagonal elements of the matrix \mathbf{X} and with the Cholesky decomposition [31]

$$\mathbf{R}_n^{-1} = \mathbf{L}^{*T} \mathbf{L} \quad (29)$$

where \mathbf{L} is an upper triangular matrix

$$\begin{aligned} \mathbf{L} &= (L_{i,j}), \quad L_{i,j} = 0 \quad \forall i > j, \\ i, j &= 1, \dots, K_a \cdot (N \cdot Q + W - 1) \end{aligned} \quad (30)$$

the continuous-valued output of the DMF for CRAD is given by

$$\hat{d}_c = \mathbf{A}^{*T} \mathbf{R}_n^{-1} \mathbf{e} \quad (31a)$$

$$= (\mathbf{L}\mathbf{A})^{*T} \mathbf{L}\mathbf{e}. \quad (31b)$$

$$= \underbrace{\text{Diag}\langle [(\mathbf{L}\mathbf{A})^{*T} \mathbf{L}\mathbf{A}]_{i,i} \rangle \mathbf{d}}_{\text{useful part}} + \underbrace{[(\mathbf{L}\mathbf{A})^{*T} \mathbf{L}\mathbf{A} - \text{Diag}\langle [(\mathbf{L}\mathbf{A})^{*T} \mathbf{L}\mathbf{A}]_{i,i} \rangle] \mathbf{d}}_{\text{ISI and MAI}} + \quad (31c)$$

$$\underbrace{(\mathbf{L}\mathbf{A})^{*T} \mathbf{L}\mathbf{n}}_{\text{perturbation}}$$

where $[X]_{i,j}$ designates the element in the i th row and the j th column of the matrix X . According to (31), the estimate \hat{d}_c contains ISI and MAI as well as a perturbation term associated with filtered noise. It follows from (31b) that S is the $K \cdot N \times K \cdot N$ -identity matrix and M is represented by

$$M = \underbrace{(\mathbf{L}\mathbf{A})^{*T}}_{\text{matched filter}} \quad \underbrace{(\mathbf{L})}_{\text{decorrelating filter}}. \quad (32)$$

The operation $\mathbf{L}\mathbf{e}$ decorrelates, i.e. prewhitens, the noise. Therefore, \mathbf{L} is a prewhitening or decorrelating filter (DF). The signal $\mathbf{L}\mathbf{e}$ is fed into the filter $(\mathbf{L}\mathbf{A})^{*T}$ which is matched to the concatenation of the $K \cdot K_a$ discrete-time channels with the impulse responses $\mathbf{b}^{(k,k_a)}$ of (11) and the DF \mathbf{L} . The concatenation of \mathbf{L} with $(\mathbf{L}\mathbf{A})^{*T}$ is a DMF.

For convenience, the Hermitian matrix

$$\mathbf{E} = \mathbf{A}^{*T} \mathbf{R}_n^{-1} \mathbf{A} = (\mathbf{L}\mathbf{A})^{*T} \mathbf{L}\mathbf{A} \quad (33)$$

is introduced. Under the assumption that the data symbols $d_n^{(k)}$ are samples of a stationary process with covariance matrix [30]

$$\mathbf{R}_d = E\{\mathbf{d}\mathbf{d}^{*T}\} \quad (34)$$

the SNR $\gamma(k,n)$ at the output of the DMF introduced in (28) is given by the equation shown at the bottom of the page (35).

The SNR $\gamma(k,n)$ introduced in (35) is maximal when neither ISI nor MAI are present. In this case, (35) reduces to

$$\gamma(k,n) = E\{|d_n^{(k)}|^2\} \cdot [\mathbf{E}]_{N \cdot (k-1)+n, N \cdot (k-1)+n} \quad k = 1, \dots, K, n = 1, \dots, N. \quad (35)$$

The conventional JD technique represented by the DMF is only suitable in the case where both ISI and MAI are negligible, as it is the case in spread spectrum multiple access (SSMA) radio systems [1] with low spectral efficiency. However, both ISI and MAI are usually considerable in mobile radio systems thus leading to impairments. In the following sections, JD techniques with considerable performance improvements over the DMF are presented.

C. Decorrelating Matched Filter Block Decision Feedback Equalizer

In this section, a decision feedback version of the DMF for CRAD termed DMF-BDFE for CRAD shall be derived setting out from the DMF for CRAD discussed in the preceding section. The DMF-BDFE for CRAD is based on interference cancellation, cf. e.g. [10], [22]. Its basic principle is presented in the following. For convenience, (31a) is displayed using the elements of the vectors \hat{d}_c and \mathbf{e} as well as of the matrix M of (32) in the following way:

$$\hat{d}_{c,n} = \sum_{j=1}^{K_a \cdot (N \cdot Q + W - 1)} M_{n,j} e_j, \quad n = 1, \dots, K \cdot N. \quad (37)$$

According to (37), the estimation of \hat{d}_c can be carried out recursively with n being the label of the recursion steps. In the following, it is assumed that the recursion starts with n equal to $K \cdot N$ and ends with n equal to one. Therefore, $\hat{d}_{c,K \cdot N}$ is determined during the first recursion step with label n equal to $K \cdot N$. Based on $\hat{d}_{c,K \cdot N}$, the discrete-valued version $\hat{d}_{q,K \cdot N}$ is generated by quantizing $\hat{d}_{c,K \cdot N}$. The effect of those parts contained in the received vector \mathbf{e} which are dependent on the data symbol $d_{K \cdot N}$ can now be reduced because the estimate $\hat{d}_{q,K \cdot N}$ of $d_{K \cdot N}$ is known. Hence, $\hat{d}_{q,K \cdot N}$ is modulated at the receiver, i.e.

$$A_{j,K \cdot N} \hat{d}_{q,K \cdot N}, \quad j = 1, \dots, K_a \cdot (N \cdot Q + W - 1) \quad (38)$$

is generated, and then (38) is subtracted from the j th component e_j , $j = 1, \dots, K_a \cdot (N \cdot Q + W - 1)$, of the received vector \mathbf{e} at the DMF-BDFE input. In the next recursion step with label n equal to $(K \cdot N - 1)$, $\hat{d}_{q,K \cdot N-1}$ is determined.

Now, the effect of $d_{K \cdot N}$ and $d_{K \cdot N-1}$ on the received vector \mathbf{e} is reduced by subtracting the modulated versions

$$\sum_{i=K \cdot N-1}^{K \cdot N} A_{j,i} \hat{d}_{q,i} = A_{j,K \cdot N-1} \hat{d}_{q,K \cdot N-1} + A_{j,K \cdot N} \hat{d}_{q,K \cdot N}, \quad j = 1, \dots, K_a \cdot (N \cdot Q + W - 1) \quad (39)$$

of $\hat{d}_{q,K \cdot N-1}$ and $\hat{d}_{q,K \cdot N}$ from the j th component e_j , $j = 1, \dots, K_a \cdot (N \cdot Q + W - 1)$, of \mathbf{e} at the DMF-BDFE input. The described procedure is continued until \hat{d}_q is entirely determined. Assuming that n is decremented from $K \cdot N$ down to one, the mathematical representation of this procedure is given by

$$\hat{d}_{c,n} = \sum_{j=1}^{K_a \cdot (N \cdot Q + W - 1)} M_{n,j} \left(e_j - \sum_{i=n+1}^{K \cdot N} A_{j,i} \hat{d}_{q,i} \right), \quad n = 1, \dots, K \cdot N. \quad (40)$$

Introducing the notation $\text{Tri}\langle X \rangle$ for the strict upper triangular matrix with vanishing diagonal elements consisting of

$$\gamma(k,n) = \frac{E\{|d_n^{(k)}|^2\} \cdot ([\mathbf{E}]_{\nu,\nu})^2}{[\mathbf{E}\mathbf{R}_d\mathbf{E}]_{\nu,\nu} + (1 - 2\text{Re}\{[\mathbf{E}\mathbf{R}_d]_{\nu,\nu}\}) \cdot [\mathbf{E}]_{\nu,\nu} + E\{|d_n^{(k)}|^2\} \cdot ([\mathbf{E}]_{\nu,\nu})^2}, \quad \nu = N \cdot (k-1) + n, \quad k = 1, \dots, K, \quad n = 1, \dots, N. \quad (35)$$

the upper triangular part of the matrix \mathbf{X} and introducing the definitions

$$[\mathbf{x}]_j^i \stackrel{\text{def}}{=} (x_i, x_{i+1}, \dots, x_j)^T \quad i \leq j. \quad (41a)$$

$$[\mathbf{X}]_{q,j}^{p,i} \stackrel{\text{def}}{=} \begin{pmatrix} X_{p,i} & X_{p,i+1} & \cdots & X_{p,j} \\ X_{p+1,i} & X_{p+1,i+1} & \cdots & X_{p+1,j} \\ \vdots & \vdots & \cdots & \vdots \\ X_{q,i} & X_{q,i+1} & \cdots & X_{q,j} \end{pmatrix} \quad p \leq q, i \leq j \quad (41b)$$

(40) can be displayed as

$$\begin{aligned} & ([\mathbf{I}]_{K \cdot N, n}^{1,1} \cdot [\text{Tri}(\mathbf{E})]_{K \cdot N, K \cdot N}^{1, n+1}) \cdot ([\hat{\mathbf{d}}_c]_n^1)^T \cdot ([\hat{\mathbf{d}}_q]_{K \cdot N}^{n+1})^T \\ & = \mathbf{A}^{*T} \mathbf{R}_n^{-1} \mathbf{e} \end{aligned} \quad (42)$$

with \mathbf{I} denoting the $K \cdot N \times K \cdot N$ -identity-matrix. The matrix $[\text{Tri}(\mathbf{E})]_{K \cdot N, K \cdot N}^{1, n+1}$ represents the feedback operator. With

$$\mathbf{M} = \mathbf{A}^{*T} \mathbf{R}_n^{-1}. \quad (43a)$$

$$\mathbf{S} = ([\mathbf{I}]_{K \cdot N, n}^{1,1} \cdot [\text{Tri}(\mathbf{E})]_{K \cdot N, K \cdot N}^{1, n+1}). \quad (43b)$$

$$\hat{\mathbf{d}} = ([\hat{\mathbf{d}}_c]_n^1)^T \cdot ([\hat{\mathbf{d}}_q]_{K \cdot N}^{n+1})^T \quad (43c)$$

The structure of (42) is recognized. Furthermore, the DMF-BDFE obviously contains a DMF, cf. (43a). However, both \mathbf{S} and $\hat{\mathbf{d}}$ given by (43b) and (43c), respectively, must be updated every time instant nT_s with n decreasing from $K \cdot N$ down to one. The signal processing expense of the presented DMF-BDFE is only moderately increased compared with the DMF presented in the preceding section.

With the definition

$$\mathbf{F} \stackrel{\text{def}}{=} \mathbf{E} - \text{Tri}(\mathbf{E}). \quad (44)$$

The SNR $\gamma(k, n)$ at the output of the DMF-BDFE is given by the equation at the bottom of the page assuming that all the past decisions which are fed back are correct. The SNR $\gamma(k, n)$ of (45) is generally lower than $\gamma(k, n)$ at the output of the DMF given by (36). Error propagation may occur in the case of incorrect past decisions. Nevertheless, the performance of the DMF-BDFE can be improved by applying channel sorting similar to [3], (53).

D. Zero Forcing Block Linear Equalizer

As mentioned in Section I, a straightforward JD technique is represented by the ZF-BLE [3], [4], [24] which is based on the Gauss-Markov estimation [23]. The ZF-BLE was discussed in [3], [4], [24] for the case of a single antenna receiver. In this section, the extension of the ZF-BLE to CRAD is demonstrated by considering \mathbf{A} of (18), \mathbf{n} according to (19) and \mathbf{e} defined by (22).

In analogy to the single antenna receiver case [3], [4], [24], the ZF-BLE for CRAD minimizes the quadratic form

$$Q(\mathbf{d}_c) = (\mathbf{e} - \mathbf{A}\mathbf{d}_c)^{*T} \mathbf{R}_n^{-1} (\mathbf{e} - \mathbf{A}\mathbf{d}_c) \quad (46)$$

where \mathbf{d}_c is a data vector with continuous valued elements $d_{c,n}$, $n = 1, \dots, K \cdot N$. The minimum of $Q(\mathbf{d}_c)$ is associated with the continuous valued and unbiased estimate

$$\begin{aligned} \hat{\mathbf{d}}_c &= (\mathbf{A}^{*T} \mathbf{R}_n^{-1} \mathbf{A})^{-1} \mathbf{A}^{*T} \mathbf{R}_n^{-1} \mathbf{e} \\ &= \mathbf{d} + \underbrace{(\mathbf{A}^{*T} \mathbf{R}_n^{-1} \mathbf{A})^{-1} \mathbf{A}^{*T} \mathbf{R}_n^{-1} \mathbf{n}}_{\text{perturbation}} \end{aligned} \quad (47)$$

of \mathbf{d} given by (14). According to (47), the estimate $\hat{\mathbf{d}}_c$ is free of ISI and MAI but still contains a perturbation term representing filtered noise.

It follows from (47) that \mathbf{S} is the $K \cdot N \times K \cdot N$ -identity matrix and \mathbf{M} can be displayed as

$$\mathbf{M} = (\mathbf{A}^{*T} \mathbf{R}_n^{-1} \mathbf{A})^{-1} \mathbf{A}^{*T} \mathbf{R}_n^{-1}. \quad (48)$$

In analogy to [3], \mathbf{M} can be further developed. With the Cholesky decomposition [31]

$$\mathbf{A}^{*T} \mathbf{R}_n^{-1} \mathbf{A} = \mathbf{H}^{*T} \Sigma^2 \mathbf{H} \quad (49)$$

where \mathbf{H} is a unit upper triangular matrix and Σ is a diagonal matrix

$$\begin{aligned} \mathbf{H} &= (H_{i,j}), \quad H_{i,j} = 0 \quad \forall i > j, \quad H_{i,i} = 1 \quad \forall i, \\ i, j &= 1, \dots, K \cdot N. \end{aligned} \quad (50a)$$

$$\Sigma = \text{Diag}(\sigma_{i,i}), \quad \sigma_{i,i} \in \mathbb{R}, \quad i = 1, \dots, K \cdot N. \quad (50b)$$

(48) can be displayed as

$$\begin{aligned} \mathbf{M} &= \underbrace{(\Sigma \mathbf{H})^{-1}}_{\text{ISI and MAI canceller}} \quad \underbrace{(\mathbf{H}^{*T} \Sigma)^{-1}}_{\text{whitening filter}} \quad \underbrace{(\mathbf{L} \mathbf{A})^{*T}}_{\text{matched filter}} \\ &\times \underbrace{(\mathbf{L})}_{\text{decorrelating filter}}. \end{aligned} \quad (51)$$

As mentioned in Section III-A, the ZF-BLE for CRAD contains a DMF which is applied to the received vector \mathbf{e} of (22). The output of the DMF is fed into the whitening filter (WF) $(\mathbf{H}^{*T} \Sigma)^{-1}$. The combination of the filters \mathbf{L} , $(\mathbf{L} \mathbf{A})^{*T}$, and $(\mathbf{H}^{*T} \Sigma)^{-1}$ shall be termed decorrelating whitened matched filter (DWMF). At the output of the DWMF, a maximum likelihood sequence estimator (MLSE) can be applied. However, the expense of such a MLSE is prohibitively high. Therefore, using a linear ISI and MAI canceller such as $(\Sigma \mathbf{H})^{-1}$ is more favorable.

$$\begin{aligned} \gamma(k, n) &= \frac{E\{|d_n^{(k)}|^2\} \cdot ([\mathbf{E}]_{\nu, \nu})^2}{[\mathbf{F} \mathbf{R}_d \mathbf{F}^{*T}]_{\nu, \nu} - (1 - 2\text{Re}\{[\mathbf{F} \mathbf{R}_d]_{\nu, \nu}\}) \cdot [\mathbf{E}]_{\nu, \nu} + E\{|d_n^{(k)}|^2\} \cdot ([\mathbf{E}]_{\nu, \nu})^2}, \\ \nu &= N \cdot (k - 1) + n, \quad k = 1, \dots, K, \quad n = 1, \dots, N \end{aligned} \quad (45)$$

With \mathbf{E} defined by (33), the SNR $\gamma(k, n)$ at the output of the ZF-BLE can be displayed as

$$\gamma(k, n) = \frac{E\{|d_n^{(k)}|^2\}}{[E^{-1}]_{N \cdot (k-1)+n, N \cdot (k-1)+n}}, \quad k = 1, \dots, K, \quad n = 1, \dots, N, \quad K, N \in \mathbb{N} \quad (52)$$

which is in general lower than $\gamma(k, n)$ of (36) at the output of the DMF. However, the ZF-BLE performs better than the DMF when ISI and MAI are considerable.

E. Zero Forcing Block Decision Feedback Equalizer

In this section, the extension of the ZF-BDFE [3] to CRAD is demonstrated. The ZF-BDFE for CRAD shall be derived setting out from the ZF-BLE for CRAD discussed in the preceding section. The structure of the unit upper triangular matrix \mathbf{H} can be exploited for establishing the ZF-BDFE. With (47) and (51), the equality

$$\begin{aligned} \mathbf{H}\hat{\mathbf{d}}_c &= \hat{\mathbf{d}}_c + (\mathbf{H} - \mathbf{I})\hat{\mathbf{d}}_c = \Sigma^{-1}(\mathbf{H}^* \Sigma)^{-1}(\mathbf{L}\mathbf{A})^{*T}\mathbf{L}\mathbf{e} \\ &= \mathbf{M}\mathbf{e} \end{aligned} \quad (53a)$$

$$(53b)$$

holds, where the matrix

$$\mathbf{M} = \Sigma^{-1}(\mathbf{H}^* \Sigma)^{-1}(\mathbf{L}\mathbf{A})^{*T}\mathbf{L} \quad (54)$$

is used and \mathbf{I} denotes the $K \cdot N \times K \cdot N$ -identity-matrix. According to (54), the ZF-BDFE contains a DWMF. From (53b), the identities

$$\hat{d}_{c, K \cdot N} = \sum_{j=1}^{K_a \cdot (N \cdot Q + W - 1)} M_{K \cdot N, j} e_j, \quad (55a)$$

$$\begin{aligned} \hat{d}_{c, n} + \sum_{j=n+1}^{K \cdot N} H_{n, j} \hat{d}_{c, j} &= \sum_{j=1}^{K_a \cdot (N \cdot Q + W - 1)} M_{n, j} e_j, \\ n &= 1, \dots, K \cdot N - 1 \end{aligned} \quad (55b)$$

follow. According to (55a), the estimate $\hat{d}_{c, K \cdot N}$ is determined by the linear superposition of the $K_a \cdot (N \cdot Q + W - 1)$ weighted elements $M_{K \cdot N, j} e_j$. All the other estimates $\hat{d}_{c, n}$, are moreover influenced by the weighted estimates $H_{n, n+1} \hat{d}_{c, n+1}, H_{n, n+2} \hat{d}_{c, n+2} \dots H_{n, K \cdot N} \hat{d}_{c, K \cdot N}$, cf. (55b). Under the assumption that the set of equations given by (55a) and (55b) is solved recursively with n descending from $K \cdot N$ down to one, a ZF-BDFE can be realized. Replacing $\hat{d}_{c, j}, j = (n+1) \dots (K \cdot N)$, in (55b) by the quantized versions \hat{d}_q, j introduced in Section III-A yields the ZF-BDFE with

$$\hat{d}_{c, K \cdot N} = \sum_{j=1}^{K_a \cdot (N \cdot Q + W - 1)} M_{K \cdot N, j} e_j, \quad (56a)$$

$$\begin{aligned} \hat{d}_{c, n} + \sum_{j=n+1}^{K \cdot N} H_{n, j} \hat{d}_{q, j} &= \sum_{j=1}^{K_a \cdot (N \cdot Q + W - 1)} M_{n, j} e_j, \\ n &= 1, \dots, K \cdot N - 1. \end{aligned} \quad (56b)$$

With (41a) and (41b), (53b) results in

$$([\mathbf{I}]_{K \cdot N, n}^{1,1} \cdot [\mathbf{H} - \mathbf{I}]_{K \cdot N, K \cdot N}^{1, n+1}) \left(([\hat{\mathbf{d}}_c]_n^1)^T \cdot ([\hat{\mathbf{d}}_q]_{K \cdot N}^{n+1})^T \right)^T = \mathbf{M}\mathbf{e}. \quad (57)$$

The matrix $[\mathbf{H} - \mathbf{I}]_{K \cdot N, K \cdot N}^{1, n+1}$ represents the feedback operation. With

$$\mathbf{S} = ([\mathbf{I}]_{K \cdot N, n}^{1,1} \cdot [\mathbf{H} - \mathbf{I}]_{K \cdot N, K \cdot N}^{1, n+1}), \quad (58a)$$

$$\hat{\mathbf{d}} = \left(([\hat{\mathbf{d}}_c]_n^1)^T \cdot ([\hat{\mathbf{d}}_q]_{K \cdot N}^{n+1})^T \right)^T \quad (58b)$$

the structure of (24) is recognized. Both \mathbf{S} and $\hat{\mathbf{d}}$ given by (58a) and (58b), respectively, must be updated every time instant nT_s , cf. Section III-C.

The SNR $\gamma(k, n)$ at the output of the ZF-BDFE is given by

$$\gamma(k, n) = E\{|d_n^{(k)}|^2\} \cdot (\sigma_{N \cdot (k-1)+n, N \cdot (k-1)+n})^2 \quad k = 1, \dots, K, \quad n = 1, \dots, N, \quad K, N \in \mathbb{N} \quad (59)$$

assuming that all the past decisions which are fed back are correct. The SNR $\gamma(k, n)$ of (59) is generally greater than $\gamma(k, n)$ at the output of the ZF-BLE given by (52). However, in the case of incorrect past decisions, the performance of the ZF-BDFE suffers from error propagation. As already mentioned in Section III-C, the performance of the ZF-BDFE can also be improved by applying channel sorting similar to [3], (53).

F. Minimum Mean Square Error Block Linear Equalizer

In [3], the MMSE-BLE was investigated for the single antenna receiver case. In this section, the extension of the MMSE-BLE to CRAD is presented. Again, the definitions for \mathbf{A} given in (18), \mathbf{n} according to (19) and \mathbf{e} introduced in (22) are used.

In the case of the MMSE-BLE for CRAD, the quadratic form

$$Q(\hat{\mathbf{d}}_c) = E\{(\hat{\mathbf{d}}_c - \mathbf{d})^{*T}(\hat{\mathbf{d}}_c - \mathbf{d})\} \quad (60a)$$

$$= E\{\|\hat{\mathbf{d}}_c - \mathbf{d}\|^2\} \quad (60b)$$

must be minimized, cf. [3], [23], [24] for the single antenna receiver. The data vector $\hat{\mathbf{d}}_c$, consists of the continuous-valued elements $d_{c, n}, n = 1, \dots, K \cdot N$. The quadratic form $Q(\hat{\mathbf{d}}_c)$ assumes its minimum when $\hat{\mathbf{d}}_c$ is equal to the continuous-valued and unbiased estimate

$$\hat{\mathbf{d}}_c = (\mathbf{A}^{*T} \mathbf{R}_n^{-1} \mathbf{A} + \mathbf{R}_d^{-1})^{-1} \mathbf{A}^{*T} \mathbf{R}_n^{-1} \mathbf{e} \quad (61a)$$

$$= \underbrace{(\mathbf{I} + (\mathbf{R}_d \mathbf{A}^{*T} \mathbf{R}_n^{-1} \mathbf{A})^{-1})^{-1}}_{\text{Wiener filter } \mathbf{W}_0} \times \underbrace{(\mathbf{A}^{*T} \mathbf{R}_n^{-1} \mathbf{A})^{-1} \mathbf{A}^{*T} \mathbf{R}_n^{-1} \mathbf{e}}_{\text{ZF-BLE}} \quad (61b)$$

$$\begin{aligned} &= \underbrace{\text{Diag}\langle\{\mathbf{W}_0\}_{i,i}\rangle \mathbf{d}}_{\text{useful part}} + \underbrace{[\mathbf{W}_0 - \text{Diag}\langle\{\mathbf{W}_0\}_{i,i}\rangle] \mathbf{d}}_{\text{ISI and MAI}} \\ &\quad + \underbrace{\mathbf{W}_0 (\mathbf{A}^{*T} \mathbf{R}_n^{-1} \mathbf{A})^{-1} \mathbf{A}^{*T} \mathbf{R}_n^{-1} \mathbf{n}}_{\text{perturbation}}. \end{aligned} \quad (61c)$$

where \mathbf{I} designates the $K \cdot N \times K \cdot N$ -identity-matrix. According to (61), the estimate $\hat{\mathbf{d}}_c$ contains a useful term an ISI and MAI term as well as a perturbation term, and it is

$$\gamma(k, n) = \frac{E\left\{\left|d_n^{(k)}\right|^2\right\} \cdot |[\mathbf{W}_0]_{\nu, \nu}|^2}{[\mathbf{W}_0 \mathbf{R}_d]_{\nu, \nu} - 2\text{Re}\{[\mathbf{W}_0 \mathbf{R}_d]_{\nu, \nu} \cdot [\mathbf{W}_0]^*_{\nu, \nu}\} + E\left\{\left|d_n^{(k)}\right|^2\right\} \cdot |[\mathbf{W}_0]_{\nu, \nu}|^2}.$$

$\nu = N \cdot (k-1) + n, \quad k = 1, \dots, K,$
 $n = 1, \dots, N, \quad K, N \in \mathbb{N}$

(64)

the output of the ZF-BLE followed by the Wiener filter [24, p. 373]

$$\mathbf{W}_0 = \left(\mathbf{I} + (\mathbf{R}_d \mathbf{A}^{*T} \mathbf{R}_n^{-1} \mathbf{A})^{-1} \right)^{-1}. \quad (62)$$

Due to (61), \mathbf{S} is identical with the $K \cdot N \times K \cdot N$ -identity matrix and \mathbf{M} can be displayed as

$$\mathbf{M} = \mathbf{W}_0 (\Sigma \mathbf{H})^{-1} (\mathbf{H}^{*T} \Sigma)^{-1} (\mathbf{L} \mathbf{A})^{*T} \mathbf{L}. \quad (63)$$

According to (63), the MMSE-BLE for CRAD contains a DWMF which is applied to the received vector \mathbf{e} of (22). The output of the DWMF is fed into the linear ISI and MAI canceller $(\Sigma \mathbf{H})^{-1}$ which is followed by the Wiener filter \mathbf{W}_0 . Since the Wiener filter minimizes the expectation of the squared norm of the estimation error vector $(\mathbf{d}_c - \mathbf{d})$, the MMSE-BLE leads to a better performance than the ZF-BLE, cf. [3] for the single antenna receiver. Furthermore, it can easily be shown that the estimation errors $(\hat{d}_{c,n} - d_n)$ and the estimated data symbols $\hat{d}_{c,n}$ are uncorrelated at the output of the MMSE-BLE for CRAD.

The SNR $\gamma(k, n)$ at the output of the MMSE-BLE is given by the equation at the top of the page (64), which is in general greater than $\gamma(k, n)$ at the output of the ZF-BLE introduced in (52).

G. Minimum Mean Square Error Block Decision Feedback Equalizer

In this section, the extension of the MMSE-BDFE [3] to CRAD is demonstrated. The MMSE-BDFE for CRAD shall be derived in a similar way to the ZF-BDFE by setting out from the MMSE-BLE for CRAD discussed in the preceding section. With the Cholesky decomposition [31]

$$\mathbf{A}^{*T} \mathbf{R}_n^{-1} \mathbf{A} + \mathbf{R}_d^{-1} = (\Sigma' \mathbf{H}')^{*T} \Sigma' \mathbf{H}' \quad (65)$$

where the matrices

$$\mathbf{H}' = (H'_{i,j}), \quad H'_{i,j} = 0 \quad \forall i > j, \quad H'_{i,i} = 1 \quad \forall i,$$

$$i, j = 1, \dots, K \cdot N. \quad (66a)$$

$$\Sigma' = \text{Diag}\{\sigma'_{i,i}\}, \sigma'_{i,i} \in \mathbb{R}, \quad i = 1, \dots, K \cdot N \quad (66b)$$

are used, and with the matrix

$$\mathbf{M} = \Sigma'^{-1} (\mathbf{H}'^{*T} \Sigma')^{-1} (\mathbf{L} \mathbf{A})^{*T} \mathbf{L} \quad (67)$$

the MMSE-BDFE can be represented as follows:

$$\begin{aligned} & \left(\mathbf{I}_{K \cdot N, n}^{1,1} \cdot [\mathbf{H}' - \mathbf{I}_{K \cdot N, K \cdot N}^{1, n+1}] \right) \left(\left([\hat{d}_c]_n^1 \right)^T \cdot \left([\hat{d}_q]_{K \cdot N}^{n+1} \right)^T \right)^T \\ & = \mathbf{M} \mathbf{e}. \end{aligned} \quad (68)$$

The matrix $[\mathbf{H}' - \mathbf{I}]_{K \cdot N, K \cdot N}^{1, n+1}$ represents the feedback operator. With

$$\mathbf{S} = \left(\mathbf{I}_{K \cdot N, n}^{1,1} \cdot [\mathbf{H}' - \mathbf{I}_{K \cdot N, K \cdot N}^{1, n+1}] \right), \quad (69a)$$

$$\hat{\mathbf{d}} = \left(\left([\hat{d}_c]_n^1 \right)^T \cdot \left([\hat{d}_q]_{K \cdot N}^{n+1} \right)^T \right)^T \quad (69b)$$

the structure of (24) is recognized.

The SNR $\gamma(k, n)$ at the output of the MMSE-BDFE is given by the equation at the bottom of the page (70), assuming that all the past decisions which are fed back are correct. The SNR $\gamma(k, n)$ of (70) is generally greater than $\gamma(k, n)$ at the output of the ZF-BDFE given by (59).

IV. SIMULATION RESULTS

In this section, the average uncoded bit error probabilities P_e obtained with the six JD techniques introduced in Section III are presented for K_a equal to one and K_a equal to two receiver antennas in the case of transmission over rural area, typical urban and bad urban mobile radio channels. In the simulations, a user bandwidth of 2 MHz is chosen and the following parameters are used:

$$K = 8, \quad (71a)$$

$$N = 20, \quad (71b)$$

$$\gamma(k, n) = \frac{E\left\{\left|d_n^{(k)}\right|^2\right\} \cdot \left|(\sigma'_{\nu, \nu})^2 - [(\mathbf{H}' \mathbf{R}_d)^{-1}]_{\nu, \nu}\right|^2}{E\left\{\left|d_n^{(k)}\right|^2\right\} \cdot \left|[[(\mathbf{H}' \mathbf{R}_d)^{-1}]_{\nu, \nu}\right|^2 - 2\text{Re}\left\{[(\mathbf{H}' \mathbf{R}_d)^{-1}]_{\nu, \nu}\right\} + (\sigma'_{\nu, \nu})^2}$$

$\nu = N \cdot (k-1) + n, \quad k = 1, \dots, K, \quad n = 1, \dots, N.$

$$K, N \in \mathbb{N} \quad (70)$$

$$Q = 14. \quad (71c)$$

$$W = 2 \text{ (rural area).} \quad (71d)$$

$$W = 11 \text{ (typical urban),} \quad (71e)$$

$$W = 21 \text{ (bad urban),} \quad (71f)$$

$$T_c = 0.5 \text{ mus.} \quad (71g)$$

The modulation scheme is 4PSK. The maximal data rate $1/T_s$ is approximately 143 kbit/s. The randomly generated signature sequences $\mathbf{c}^{(k)}$, $k = 1, \dots, 8$, are binary.

Variations of the received power due to path losses and lognormal fading are assumed to be eliminated by applying a perfect power control. However, the variations of the received power due to Rayleigh fading are present in the received signals. In the simulations,

$$\mathbf{R}_d = \mathbf{I} \quad (72)$$

corresponding to

$$E\left\{\left|d_n^{(k)}\right|^2\right\} = 1 \quad (73)$$

is chosen. The additive noise is assumed to be white and normally distributed with covariance matrix

$$\mathbf{R}_n = \sigma^2 \cdot \mathbf{I}. \quad (74)$$

Therefore, the SNR per bit

$$\gamma_b(k, n) = \gamma_b(k) = \sum_{k_a=1}^{K_a} \frac{\|\mathbf{b}^{(k, k_a)}\|^2}{\sigma^2}, \quad k = 1, \dots, K, \quad n = 1, \dots, N, \quad K, K_a, N \in \mathbb{N} \quad (75)$$

at the input of the data detector is independent of n . With the SNR per bit $\gamma(k, n)$ at the output of the data detector operating with a particular JD technique presented in Section III, with the well-known bit error probability $P_b(\gamma(k, n))$ for 4PSK for signaling over an additive Gaussian noise channel as a function of the SNR per bit $\gamma(k, n)$ [2] and with the density function $p(\gamma(k, n))$ of the SNR per bit $\gamma(k, n)$ [2] resulting from the varying combined channel impulse responses $\mathbf{b}^{(k, k_a)}$, the average uncoded bit error probability is given by

$$P_e = \frac{1}{K \cdot N} \cdot \sum_{k=1}^K \sum_{n=1}^N \int_0^\infty P_b(\gamma(k, n)) \cdot p(\gamma(k, n)) d(\gamma(k, n)) \quad (76)$$

[4]. In the case of a fixed choice of $\mathbf{b}^{(k, k_a)}$ and σ^2 , $P_b(\gamma(k, n))$ can be determined. However, the variations of $\mathbf{b}^{(k, k_a)}$ must be simulated, thus allowing the calculation of P_e according to (76) [4].

The authors intend to determine lower bounds on the error rate performance and thus assume that all the $K \cdot K_a$ combined channel impulse responses $\mathbf{b}^{(k, k_a)}$ are linearly independent ("uncorrelated"). In practice the $K \cdot K_a$ radio channels can, for instance, be made uncorrelated by choosing sufficiently large spacings of the receiver antennas, cf. [33, ch. 9] for a detailed discussion. However, even close proximity of the receiver antennas leads to uncorrelated radio channels [17] resulting

from antenna coupling effects. Nevertheless, the present concept of JD with CRAD is not restricted to the case of two receiver antennas, which are widely spread, but can also be applied to antenna configurations consisting of many closely spaced antenna elements for which case correlations between the received signals occur. Detailed investigations concerning the effects of antenna spacing and antenna configuration on the correlation properties of the received signals, and thus on the data detection by using JD techniques, shall be carried out in the future.

In Fig. 4-6, P_e achievable with the six JD techniques of Sections III-B–F are shown versus the average SNR per bit and per antenna $\bar{\gamma}_b/K_a$ obtained by averaging over the SNR's $\gamma_b(k)$ of (75) under the assumptions that

- σ^2 and $\mathbf{b}^{(k, k_a)}$, $k = 1, \dots, K$, $k_a = 1, \dots, K_a$, are perfectly known at the receiver and
- no error propagation occurs in the case of the DMF–BDFE, the ZF–BDFE and the MMSE–BDFE.

In the case of error propagation, which shall be studied in the near future, a moderate SNR degradation is expected.

It follows from Fig. 4-6 that for the considered channel models and for one, as well as two receiver antennas, the MMSE–BDFE performs best followed by the ZF–BDFE, the MMSE–BLE and the ZF–BLE. The average uncoded bit error probabilities P_e associated with the DMF for $\gamma(k, n)$ of (36) cannot be reached by the MMSE–BDFE, the ZF–BDFE, the MMSE–BLE, and the ZF–BLE. Therefore, the DMF for $\gamma(k, n)$ of (36) represents a lower bound to P_e of the considered suboptimum JD techniques. The average uncoded bit error probability P_e of the MMSE–BLE always merges into P_e of the DMF for $\bar{\gamma}_b/K_a \rightarrow 0$ and into P_e of the ZF–BLE for $\bar{\gamma}_b/K_a \rightarrow \infty$, cf. (31a), (47), and (61a). The same statement is valid for the decision feedback versions of the ZF and the MMSE equalizers.

As expected, P_e obtained with the DMF for $\gamma(k, n)$ of (35) and with the DMF–BDFE for $\gamma(k, n)$ of (45) merge into irreducible error floors because neither ISI nor MAI are fully removed. According to Fig. 4-6, the magnitudes of these irreducible error floors are dependent on both the considered channel model as well as the number K_a of receiver antennas. The effect of the channel model on the magnitudes of the irreducible error floors is rather small whereas the magnitudes of the irreducible error floors are significantly decreased when K_a is increased. Since the performance of a DMF and a DMF–BDFE is limited to such irreducible error floors, neither DMF nor DMF–BDFE are well suited for JD in interference limited scenarios. Nevertheless, an acceptable error performance is obtainable when powerful forward error correction (FEC) coding is applied in combination with the DMF for CRAD or the DMF–BDFE for CRAD in the case of K_a equal to two receiver antennas.

The application of two receiver antennas instead of a single receiver antenna generally leads to a performance improvement of ≈ 3 dB because the unperturbed parts contained in \mathbf{e} received over the two receiver antennas are superimposed coherently whereas the perturbations contained in \mathbf{e} are added noncoherently at the data detector. The additional performance

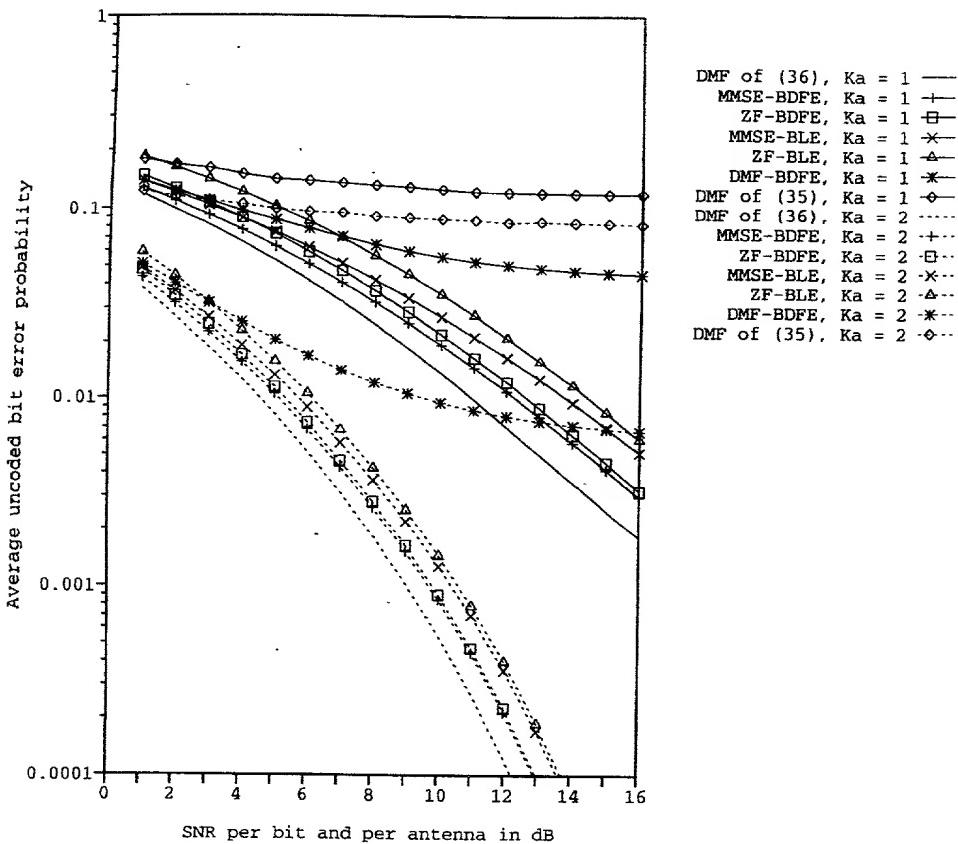


Fig. 4: Average uncoded bit error probabilities P_e versus the average SNR per bit and per antenna $\bar{\gamma}_b/K_a$ for transmission over rural area radio channels when DMF, DMF-BDFE, ZF-BLE, MMSE-BLE, ZF-BDFE, and MMSE-BDFE are used for JD in the case of $K_a = 1.2$.

improvements due to the combating of Rayleigh fading can be determined by comparing the average uncoded bit error probabilities P_e of the DMF for $\gamma(k, n)$ of (36) achievable for a single antenna receiver as well as for CRAD. The following additional performance improvements due to the reduction of the impairing effects because of Rayleigh fading are deduced from Fig. 4-6:

$$\begin{aligned} \text{additional performance improvement} &\approx 3.5 \text{ dB}@P_e \\ &= 10^{-2} \text{ (rural area)}, \end{aligned} \quad (77a)$$

$$\begin{aligned} \text{additional performance improvement} &\approx 1.5 \text{ dB}@P_e \\ &= 10^{-3} \text{ (typical urban)}, \end{aligned} \quad (77b)$$

$$\begin{aligned} \text{additional performance improvement} &\approx 1.25 \text{ dB}@P_e \\ &= 10^{-3} \text{ (bad urban)}. \end{aligned} \quad (77c)$$

The reason for the different values of these additional performance improvements is the varied inherent diversity potential of the considered mobile radio channels. For instance, the inherent diversity potential of the bad urban mobile radio channel is the largest. Therefore, the achievable additional performance improvement is the lowest. The following total performance improvements obtained with the DMF for $\gamma(k, n)$ of (36) result:

$$\begin{aligned} \text{total performance improvement} &\approx 6.5 \text{ dB}@P_e \\ &= 10^{-2} \text{ (rural area)}, \end{aligned} \quad (78a)$$

$$\begin{aligned} \text{total performance improvement} &\approx 4.5 \text{ dB}@P_e \\ &= 10^{-3} \text{ (typical urban)}, \end{aligned} \quad (78b)$$

$$\begin{aligned} \text{total performance improvement} &\approx 4.25 \text{ dB}@P_e \\ &= 10^{-3} \text{ (bad urban)}. \end{aligned} \quad (78c)$$

It follows from Fig. 4-6 that the total performance improvements achievable with the MMSE-BDFE, the ZF-BDFE, the MMSE-BLE, and the ZF-BLE by applying two receiver antennas are larger than the total performance improvements of (78) because the SNR degradations of the considered suboptimum JD techniques apparent in (52), (59), (64), and (70) are reduced by applying CRAD. For the same reason, the average uncoded bit error probabilities P_e achieved with the MMSE-BDFE, the ZF-BDFE, the MMSE-BLE, and the ZF-BLE when CRAD is considered do not differ as much as in the case of a single antenna receiver.

As an example, the average uncoded bit error probabilities P_e achievable with the ZF-BLE and the ZF-BDFE in bad urban areas are discussed, cf. Fig. 6. In order to obtain P_e equal to 10^{-3} , $\bar{\gamma}_b/K_a$ approximately equal to 12.7 dB is required in the case of the ZF-BLE for a single antenna receiver. However, the required $\bar{\gamma}_b/K_a$ is only about 6.3 dB in the case of the ZF-BLE when two receiver antennas are applied thus leading to a total performance improvement of about 6.4 dB. Therefore, the total performance improvement of the ZF-BLE exceeds the total performance improvement of (78c) by about 2.15 dB. When the ZF-BDFE is used P_e equal to 10^{-3} can already be reached for $\bar{\gamma}_b/K_a$ approximately equal to 11 dB in the case of a single antenna receiver whereas $\bar{\gamma}_b/K_a$ of about 5.6 dB is necessary when two receiver antennas are applied. The total performance improvement of the ZF-BDFE equal to 5.4 dB therefore exceeds the total performance improvement of (78c) by about 1.15 dB.

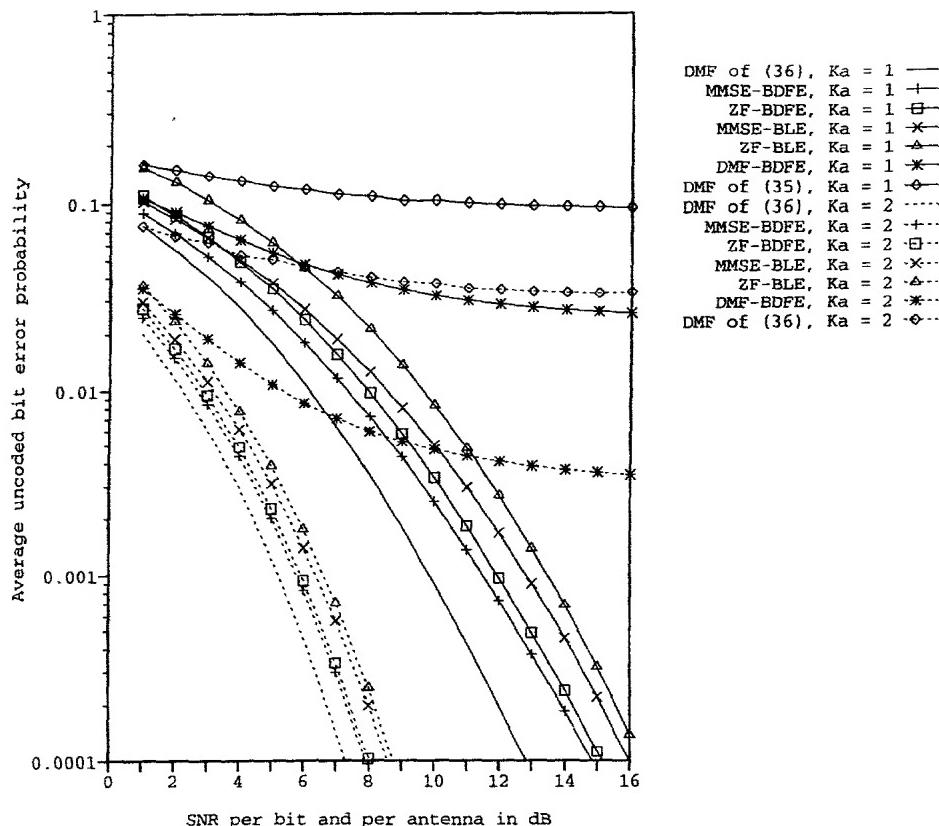


Fig. 5. Average uncoded bit error probabilities P_b versus the average SNR per bit and per antenna $\bar{\gamma}_b/K_n$ for transmission over typical urban radio channels when DMF, DMF-BDFE, ZF-BLE, MMSE-BLE, ZF-BDFE, and MMSE-BDFE are used for JD in the case of $K_n = 1.2$.

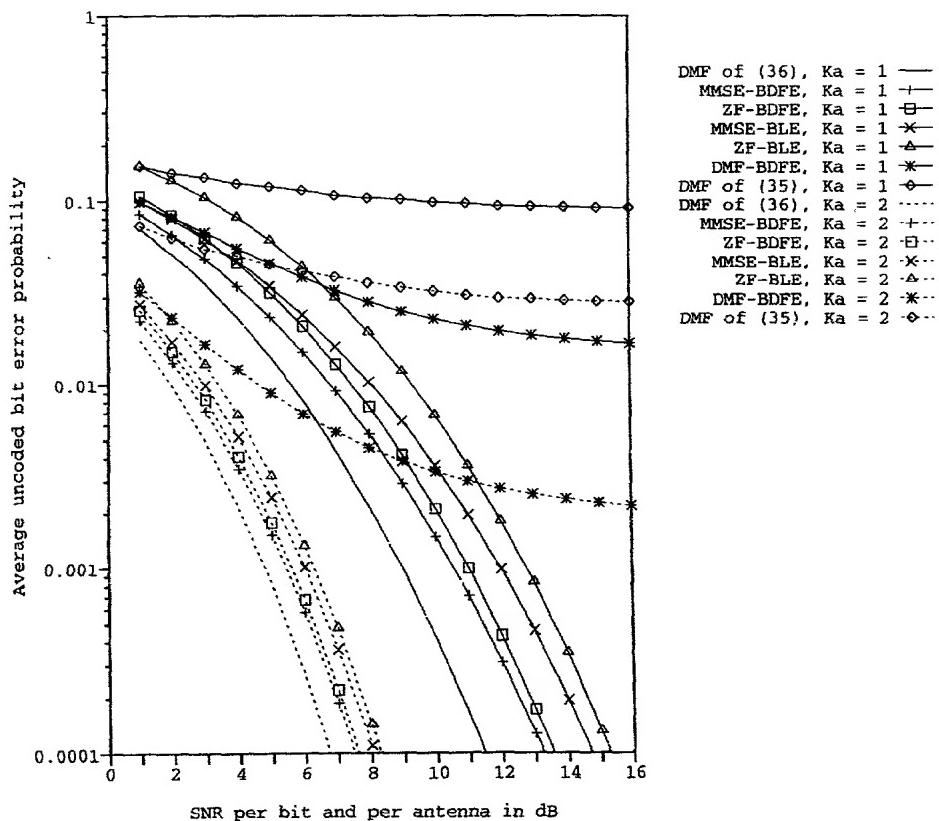


Fig. 6. Average uncoded bit error probabilities P_b versus the average SNR per bit and per antenna $\bar{\gamma}_b/K_n$ for transmission over bad urban radio channels when DMF, DMF-BDFE, ZF-BLE, MMSE-BLE, ZF-BDFE, and MMSE-BDFE are used for JD in the case of $K_n = 1.2$.

V. CONCLUSION

In the present paper, six well-known JD techniques, namely the DMF, the DMF-BDFE, the ZF-BLE, the ZF-BDFE, the MMSE-BLE, and the MMSE-BDFE, applicable to synchronous JD-CDMA mobile radio systems have been extended to CRAD and their viability has been shown by simulations. Since the MMSE equalizers minimize the expectation of the squared norm of the estimation error vector ($d_e - d$), they perform better than the ZF equalizers. Furthermore, the decision feedback versions of the DMF, the ZF, and the MMSE equalizers perform better than the corresponding linear versions. In addition, the application of CRAD yields considerable performance improvements over the single antenna receivers especially for mobile radio channels with low inherent diversity potentials.

ACKNOWLEDGMENT

The support of the Regionales Hochschulrechenzentrum Kaiserslautern (RHRK), especially of the VP100 supercomputer administrators Dr. Reinhard Corr and Dr. Tonnis Pool, is gratefully acknowledged by the authors. Furthermore, the authors are deeply indebted to Dipl.-Ing. Anja Klein, Dipl.-Ing. Markus Naßhan, Dipl.-Ing. Bernd Steiner, and Prof. Dr.-Ing. habil. Paul Walter Baier for invaluable discussions and support during the preparation of the presented results as well as the transfer of FORTRAN77 source code which eased the development of the simulation programs.

REFERENCES

- [1] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spread Spectrum Communications*, vol. III. Rockville, MD: Computer Science Press, 1985.
- [2] J. G. Proakis, *Digital Communications*, 2nd ed. New York: McGraw-Hill, 1989.
- [3] X. Lasne, C. Baroux, and G. K. Kaleh, "Joint reception of multi-user data for synchronous code-division multiple access," COST 231 Tech. Doc. (92)85, Helsinki, Finland, 1992.
- [4] A. Klein and P. W. Baier, "Linear unbiased data estimation in mobile radio systems applying CDMA," *IEEE J. Select. Areas Commun.*, vol. 11, pp. 1058-1066, 1993.
- [5] W. C. Y. Lee, "Overview of cellular CDMA," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 291-302, 1991.
- [6] A. Baier, "Multi-rate DS-CDMA: A promising access technique for third-generation mobile radio systems," in *Proc. 4th Int. Symp. Personal, Indoor, and Mobile Radio Commun. (PIMRC '93)*, Yokohama, Japan, 1993, pp. 114-118.
- [7] K. S. Gilhousen, I. M. Jacobs, R. Padovani, A. J. Viterbi, L. A. Weaver, and C. E. Wheatley, "On the capacity of a cellular CDMA system," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 303-312, 1991.
- [8] S. Verdú, "Minimum probability of error for asynchronous Gaussian multiple-access channels," *IEEE Trans. Info. Theory*, vol. 35, pp. 123-136, 1989.
- [9] R. Lupas and S. Verdú, "Linear multiuser detectors for synchronous code-division multiple access channels," *IEEE Trans. Info. Theory*, vol. 35, pp. 123-136, 1989.
- [10] R. S. Mowbray, R. D. Pringle, and P. M. Grant, "Increased CDMA system capacity through adaptive cochannel interference regeneration and cancellation," *Inst. Elect. Eng. Proc.*, vol. 139, pt. I, pp. 515-524, 1992.
- [11] G. K. Kaleh, "Zero-forcing decision-feedback equalizer for packet data transmission," in *Proc. IEEE Int. Conf. Commun. (ICC '93)*, Geneva, Switzerland, 1993, pp. 1752-1756.
- [12] M. Siala and G. K. Kaleh, "Equalization for orthogonal frequency division multiplexing systems," in *Proc. 43rd IEEE Vehicular Technol. Conf. (VTC '93)*, Secaucus, NJ, 1993, pp. 649-652.
- [13] A. Kajiwara and M. Nakagawa, "Crosscorrelation cancellation in SS/DS block demodulator," *IEICE Trans. Commun.*, vol. E 74, pp. 2596-2602, 1991.
- [14] S. S. Schneider, "Optimum detection of code division multiplexed signals," *IEEE Trans. Aerospace and Electronic Syst.*, vol. AE-15, pp. 181-185, 1979.
- [15] S. U. H. Qureshi, "Adaptive equalization," *Proc. IEEE*, vol. 73, pp. 461-468, 1985.
- [16] T. Haug, "Overview of GSM: Philosophy and results," *Int. J. Wireless Info. Networks*, vol. 1, pp. 7-16, 1994.
- [17] R. W. Lorenz, "Vergleich der digitalen Mobilfunksysteme in Europa (GSM) und in Japan (JDC) unter besonderer Berücksichtigung der Wirtschaftlichkeitaspekte," *Der Fernmelde-Ingenieur*, vol. 47, no. 1&2/93, 1993.
- [18] P. A. Ramsdale, "Personal communications in the UK—Implementation of PCN using DCS 1800," *Int. J. Wireless Info. Networks*, vol. 1, pp. 29-36, 1994.
- [19] J. Blanz, A. Klein, M. Naßhan, and A. Steil, "Performance of a hybrid C/TDMA mobile radio system applying joint detection and coherent receiver antenna diversity," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 568-579, 1994.
- [20] B. Steiner and P. Jung, "Optimum and suboptimum channel estimation for the uplink of CDMA mobile radio systems with joint detection," to be published in *Eur. Trans. Telecommun. Related Technol. (ETT)*, Special Issue on "Multiple Access in Radio Communication Networks," vol. 5, pp. 39-50, 1994.
- [21] L. Turin, "Introduction to spread-spectrum antimaltipath techniques and their application to urban digital radio," *Proc. IEEE*, vol. 68, pp. 328-353, 1980.
- [22] R. Kohno, H. Imai, M. Hatori, and S. Pasupathy, "An adaptive canceller of co-channel interference for spread spectrum multiple access communication networks in a power line," *IEEE J. Select. Areas Commun.*, vol. 8, pp. 641-649, 1990.
- [23] K. Brammer and G. Siffling, *Kalman-Bitzy-Filter—Deterministische Beobachtungen und Stochastische Filterung*. Munich, West Germany: Oldenbourg, 1975.
- [24] A. D. Whalen, *Detection of Signals in Noise*. New York: Academic Press, 1971.
- [25] M. Schwartz, W. R. Bennett, and S. Stein, *Communication Systems and Techniques*. New York: McGraw-Hill, 1966.
- [26] P. Höher, "A statistical discrete-time model for the WSSUS multipath channel," *IEEE Trans. Veh. Technol.*, vol. 41, pp. 461-468, 1992.
- [27] "COST 207: Digital Land Mobile Radio Communications," final rep., Luxembourg, Office for Official Publications of the European Communities, 1989.
- [28] H. Schulze, "Stochastische Modelle und digitale Simulation von Mobilfunkanälen," *Kleinheubacher Berichte*, vol. 32, pp. 473-483, 1989.

Peter Jung (S'91-M'92) was born in 1964 in Kaiserslautern, Germany. From 1983 until 1990 he studied physics and electrical engineering at the University of Kaiserslautern, Germany, where he received the M.Sc. (Dipl.-Phys.) and the Ph.D. (Dr.-Ing.) degrees in 1990 and 1993, respectively.



From May 1990 until September 1992 he was with the Microelectronics Centre of the University of Kaiserslautern (ZMK) where he was engaged in the design and implementation of Viterbi equalizers for mobile radio applications. In October 1992 he joined the Research Group for RF Communications as a senior member of staff and senior lecturer. His present interests are in signal processing.

Dr. Jung is a member of VDE-GME, VDE-ITG, and AES.

Josef Blanz was born in 1967 in Kaiserslautern, Germany. He received the M.Sc.EE (Dipl.-Ing.) degree in 1993 from the University of Kaiserslautern.

From 1990 until 1992 he was a freelance co-worker with the Microelectronics Centre of the University of Kaiserslautern (ZMK) where he was engaged in the design of digital circuits based on field programmable gate arrays. In 1993 he joined the Research Group for RF Communications of the University of Kaiserslautern as a research engineer.

Mr. Blanz's present interests are in the field of antenna diversity techniques for mobile radio systems.

